

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 58-132849

(43)Date of publication of application : 08.08.1983

(51)Int.Cl.

G06F 15/02

G06F 3/147

(21)Application number : 58-007158

(71)Applicant : SHARP CORP

(22)Date of filing : 17.01.1983

(72)Inventor : HASHIMOTO SHINTARO

KOTANI YASUHIRO

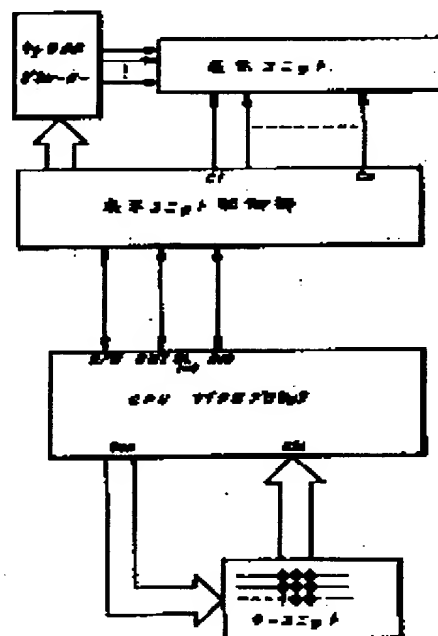
FUJIKAWA YOSHIYUKI

(54) ELECTRONIC DEVICE

(57)Abstract:

PURPOSE: To discriminate contents of a message in the message receiving side only with power-on of the message receiving side when the message is reported, by displaying automatically stored characters, symbols, etc. at the power-on time.

CONSTITUTION: Characters, symbols, etc. inputted from a key unit part are subjected to operation processings in a CPU part and are stored in an RAM of a CPU in a form of the character code, and this code is transferred to a buffer memory of a display unit controlling part. A display controlling part sends the output of the buffer memory to a character generator part and gives display information to a display unit part to display characters, symbols, etc. on a display unit part. characters, symbols, etc. exceeding the number of display digits are stored in the buffer memory, and they are displayed while being shifted successively. Contents of the memory are protected as they are even if power is turned off, and characters, symbols, etc. stored in the memory are automatically displayed successively at the power-on time, and thus, the message is inputted and presented.



べきデータの送信と可聴音データの送信とを共に実行する送信実行手段とを具えたことを特徴とする。

〔作用〕

本発明によれば、送信すべきデータの送信と可聴音のメッセージデータの送信とを共に実行することによって、確実に送信すべきデータの送信を行うことができる。

〔実施例〕

以下、図面を参照して本発明を詳細に説明する。

第1図は本発明の一実施例の構成を示す。

- 4) 同第5頁第20行および第7頁第1行の「本発明文字処理装置」を「本実施例の装置」に補正する。
- 5) 同第7頁第1行の「処理した」の次に「データ情報としての」を挿入する。
- 8) 同第14頁第14行～第19行を以下の通り補正する。

「以上説明したように、本発明によれば、送信

すべきデータの送信と可聴音のメッセージデータの送信とを共に実行することによって、確実に送信すべきデータの送信を行い得るように構成したデータ通信装置を提供することが可能となった。」

- 7) 同第15頁第1行～第7行を以下の通り補正する。

「第1図は本発明データ通信装置の一実施例を示すブロック線図、

第2図は本発明データ通信装置の一例におけるキーボードの構成配置の例をそれぞれ示す線図、

第3図は同じくそのランダムアクセスメモリの記憶の態様の例を示す線図、

第4図は同じくその文字情報伝送の態様の例を示すフローチャートである。」

以 上

別紙

特許請求の範囲

1) データの送信を指示する信号を発生する指示信号発生手段と、

前記指示信号発生手段からの指示信号に基づいて、送り先側との回線が接続されたか否かを判定する判定手段と、

前記判定手段によって前記回線の接続状態を確認した後に、送信すべきデータの送信と可聴音データの送信とを共に実行する送信実行手段とを具えたことを特徴とするデータ通信装置。

2) 前記送信すべきデータは、文字情報を表わす文字コードであることを特徴とする特許請求の範囲第1項記載のデータ通信装置。

3) 前記可聴音データは、データの転送に係わるメッセージ情報であることを特徴とする特許請求の範囲第1項または第2項記載のデータ通信装置。

2-
4-

⑨ 日本国特許庁 (JP)

⑩ 特許出願公開

⑫ 公開特許公報 (A)

昭58—132849

⑬ Int. Cl.³
G 06 F 15/02
3/147

識別記号

庁内整理番号
7343—5B
7218—5B

⑭ 公開 昭和58年(1983)8月8日

発明の数 1
審査請求 有

(全 48 頁)

⑮ 電子機器

大阪市阿倍野区長池町22番22号
シャープ株式会社内

⑯ 特 願 昭58—7158

⑰ 発 明 者 藤川嘉之

⑱ 出 願 昭54(1979)4月10日

大阪市阿倍野区長池町22番22号

⑲ 特 願 昭54—44051の分割

シャープ株式会社内

⑳ 発 明 者 橋本伸太郎

㉑ 出 願 人 シャープ株式会社

大阪市阿倍野区長池町22番22号

大阪市阿倍野区長池町22番22号

シャープ株式会社内

㉒ 代 理 人 弁理士 福士愛彦 外2名

㉓ 発 明 者 小谷泰博

明 細 書

1. 発明の名称

電子機器

2. 特許請求の範囲

1. 電源オン時に記憶している文字、記号等を自動的に表示することを特徴とする電子機器。

3. 発明の詳細な説明

本発明は、文字記号等を複数記憶し、必要な時表示する機能と計算機能を合せ持った電子装置に関するものである。

従来より、文字記号等をキーより入力し記憶させるものとして電話番号や氏名を記憶できるもの等があるが、これらは表示部の表示桁数以上の文字記号を同時に表示することができない。

これに対し本発明は、表示部の表示桁数以上の文字記号を区切ることなく記憶し、これを連続的にシフトさせながら表示することができる。即ち、記憶させる場合は設定モードにして所望の文字記号をキーより入力するが、表示桁数以上入力することができるため文章として記憶させることがで

き、またこれを表示させる場合、記憶内容をランニング表示することにより表示桁数以上の文字記号も区切ることなく連続的に表示することができる。

一方、計算機として使用する場合桁の読み誤りを防止するためスタティック（静止）表示とさせる。本発明はこのような電子機器を提供するものである。すなわち本発明は下記に列挙する電子機器を提供することを目的としている。

(1) 外部入力により文字記号等を記憶し必要に応じ表示するものであって表示桁数以上の文字記号等を記憶しこれを順次シフトさせながら表示することが可能な文字記憶電子機器。

(2) 数値を表示する表示部において、演算数或いは被演算数と共に演算記号を表示することを特徴とする電子機器。

(3) 文字記憶部内の文字記号の末尾と先頭の間に特殊記号を挿入し繰返し文字記号等の表示をする電子機器。

(4) 第1項に於て、記憶文字記号が一度に表示可

能であるか否かを検出し、スタティック表示とランニング表示を自動的に選択することを特徴とする電子機器。

(6) 第1項の文字記憶、表示手段と計算手段とを共に有するものであってキー入力部と表示部とを夫々共用したことを特徴とする電子機器。

(6) 第5項において計算途中或いは計算結果等のデータの表示はスタティック表示としたことを特徴とする電子機器。

(7) 文字記号等を記憶する手段を持ち、文字記号等の記憶状態を表示することを特徴とする電子機器。

(8) 第2項に於て演算記号は被演算数の右側に、又演算数の左側に表示することを特徴とする電子機器。

(9) 第7項に於て文字記号書き込み状態であるか否かの区別と、文字記号の記憶状態とを同一の表示部で表示することを特徴とする電子機器。

(10) 文字記号等を記憶、表示する手段と計算を行う手段とを有するものに於て、文字記号等の表示

報はディスプレイユニット制御部より供給される。すなわち、ディスプレイユニット制御部は、ディスプレイユニットに供給する表示情報をキャラクターコードの形式で保持するバッファメモリを有しており、その出力はキャラクタージェネレーターに入力され、表示用セグメント情報に変換された後、ディスプレイユニットに供給される。ディスプレイユニット制御部の出力 $C_1 \sim C_n$ はディスプレイユニットに対する制御信号であり、この信号によりディスプレイユニット制御部内のキャラクターコードはセグメント情報に変換され常時表示されている。

したがって、今何らかの表示を行いたい場合には、ディスプレイユニットの表示桁あるいは、セグメントに対応したディスプレイユニット制御部内のバッファメモリに所望のキャラクターコードあるいは、ビット情報を書き込むことにより、常にディスプレイユニットに情報が出力され表示される。ディスプレイユニット制御部内のバッファメモリは、RAMの形で構成されており、その情報を逆に読み出すこともできる。ディスプレイユニット制御部

状態で割込み計算が可能であることを特徴とする電子機器。

(11) 外部入力により文字記号等を記憶し必要に応じて表示するものであって、数字と文字を同時入力することが可能な文字記号等の記憶装置を有する電子機器。

(12) 電源ON時に記憶されている文字記号等が自動的に表示することを特徴とする電子機器。

(本発明実施例の構成)

本発明の全体構成は、第9図に示す通り、キーユニット部、ディスプレイユニット部、ディスプレイユニットに表示情報を与え制御する為のディスプレイユニット制御部、バッファ部、ディスプレイユニット制御部より表示用コードを入力し、表示用パターン情報に変換するキャラクタージェネレーター部、及びキー入力の制御、ディスプレイ制御部への表示情報の供給その他、演算処理、表示情報の処理等を行うCPU部により構成される。

ディスプレイユニットは表示媒体であり、表示情

への書き込み読み出しの制御を行う回路がCPUで構成されている。CPUから出力されるBM, BL信号は、ディスプレイユニット制御部のバッファメモリの番地を指定するアドレス信号、DIO信号はデータバス信号、R/W信号はバッファメモリへの書き込み信号である。キーユニットは、CPUよりの出力信号Son及びCPUへの入力信号Kinとの組み合わせで制御される。

ディスプレイユニット制御部内のバッファメモリは第10図の構成になっており、1アドレス4ビットとすると、 $BL=0, 1, \dots$ の各8ビットで1キャラクターコードを記憶するものとする。そしてディスプレイユニット制御部の前記メモリと全く対応したメモリを、CPU内のRAMエリアに構成させている。CPUの処理により、表示内容を準備する時はこのCPU内のRAMにキャラクターコードの形式で記憶しておき、表示する時点でこれらのコードをディスプレイユニット制御部のバッファメモリに転送する。前に説明した通り、バッファメモリに転送されたキャラクターコードは

常時キャラクタジェネレータを介してセグメント情報に変換されディスプレイユニットに与えられている。

第25図にCPUの回路ブロック構成図を示す。図における各部の記号及びその内容は次の通りである。なお①、②、……の数字はプログラム装置から指定される制御命令を表している。また下記の説明においてフリップフロップをF/Fと略記する。

RAM (ランダムアクセスメモリ) : 入出力は4ビット単位に行われ、デジットアドレスとファイルアドレスを指定することによって所望デジット内容を入出力できるメモリである。

BL : メモリRAMのデジットアドレスカウンタである。

DC₁ : メモリRAMのデジットアドレスデコーダである。

BM : メモリRAMのファイルアドレスカウン

する。

G₄ : メモリファイルアドレスBMの入力ゲートである。①の時は加算器AD₂の出力を、④はオペランドI_Aを、⑥はアキュムレータACCの内容を出力する。

G₅ : メモリRAMのファイル選択ゲートである。

DC₃ : オペランドI_Aのデコーダである。オペランドI_Aを解説し、メモリの所望ビット指定信号をゲートG₆に入力させる。

G₆ : メモリRAMの入力ゲートである。②の時はオペランドデコーダDC₃で指定されたメモリの所望ビットに2進数1を入力させ、③の時はDC₃で指定されたメモリの所望ビットに2進数0を入力させる回路を内蔵し、又、④でアキュムレータACCの内容を出力する。

ROM : リードオンリーメモリである。

PL : リードオンリーメモリROMの所望ステップを指定するプログラムカウンタであ

たである。

DC₂ : メモリRAMのファイルアドレスデコーダである。

AD₁ : 制御命令⑩が与えられた時は減算器として、⑪が与えられぬ時は加算器として動作する加減算器である。

AD₂ : 加算器である。

G₁ : 加減算器AD₁の一方の入力に数値1或いはオペランドI_Aのいずれかを与えるためのゲートである。⑤の時は1を⑥の時はI_Aを出力する。

G₂ : メモリデジットアドレスカウンタBLの入力ゲートである。⑩の時は加減算器AD₁の出力を、⑪の時はオペランドI_Aを、⑫の時はオペランドI_Bをそれぞれ出力する。

G₃ : 加算器AD₂の一方の入力に数値1、或いはオペランドI_Aのいずれかを与えるためのゲートである。⑬の時は数値1を、⑭の時はオペランドI_Aをそれぞれ出力

する。

DC₄ : リードオンリーメモリROMのステップアクセスデコーダである。

G₇ : リードオンリーメモリROMの出力ゲートである。ジャッジF/F Jがセットされた時は、ROMの出力のインストラクションデコーダDC₅への伝達が遮断される。

DC₅ : インストラクションデコーダである。

ROMからのインストラクションコードを解説するもので、ROMのインストラクションコードはオペレーションコード部分I₀とオペランド部分I_A、I_Bに分けられ、オペレーションコードを解説し、そのオペレーションコードに対応して制御命令①～⑭のいずれかを発生させる。又、オペランドをとともなうオペレーションコードであることを判断し、その時に、オペランドI_A又はI_Bをそのまま出力させる回路を内蔵する。

AD₃ : プログラムカウンタPLの内容に数値1を加え、カウントアップさせるための加算器である。

G₈ : プログラムカウンタP₂の入力ゲートである。④の時はオペランドI_Aを出力し、⑤の時はプログラムスタックレジスタSPの内容を伝達する。⑥⑦の処理時及びゲートG₃₉用の⑧の処理時は、加算器AD₃の出力は伝達されない。⑨⑩⑪以外はAD₃出力を伝達し、自動的にプログラムカウンタPLの内容に1を加える。

FC : フラッグF/Fである。

G₉ : フラッグF/F FCの入力ゲートである。④の時は2進数1を、⑤の時は2進数0をそれぞれFCに入力させる。

G₁₀ : キー信号発生ゲートである。フラッグF/F FCがリセット状態(0)の時はメモリデジットアドレスデコーダの所望出力をそのまま出力させ、フラッグF/F FCがセット状態(1)の時はDC₁出

を内蔵する。④の時はCに1を、⑤の時はCに0を入力する。

G₁₃ : キャリイを含めた2進加算を加算器AD₄で行わせるためのキャリイC入力ゲートである。⑥の時にキャリイF/F Cの出力を加算器AD₄に伝達する。

G₁₄ : 加算器AD₄の入力ゲートである。⑥の時はメモリRAMの出力を、⑦の時はオペランドI_Aを伝達する。

F : 4ビットで構成される出力バッファレジスタである。

G₁₅ : 出力バッファレジスタFの入力ゲートである。④の時にアキュムレータACCの内容を伝達し、Fに入力する。

SD : 出力デコーダである。出力バッファレジスタFの内容を解説し、表示体セグメント信号SS₁~SS_nに変換する。

W : 出力バッファレジスタである。

SHC : 出力バッファレジスタWの全ビット内容を一齐に1ビット右シフトするための出

力のいかににかかわらずI₁~I_nの出力を一齐に1にする回路を内蔵する。

ACC : 4ビットで構成されるアキュムレータである。

X : 4ビットで構成されるテンポラリレジスタである。

G₁₁ : テンポラリレジスタXの入力ゲートである。⑥の時はアキュムレータACCの内容を伝達し、⑦の時はスタックレジスタSXの内容を伝達する。

AD₄ : 加算器である。アキュムレータACCの内容と他のデータを2進加算するために用いられる。2進加算の際第4ビットの加算でキャリイが出ればC₄出力を1にする。

C : キャリイF/Fである。

G₁₂ : キャリイF/F入力ゲートである。①の時にもし第4ビットキャリイC₄が1であればキャリイF/F Cに1を入力し、C₄が0であればCに0を入力する回路

力バッファレジスタWのシフト回路である。⑥又は⑦が発生した時に動作する。

G₁₆ : 出力バッファレジスタWの入力ゲートである。⑥の時にはWの第1ビットに1を入力し、⑦の時にはWの第1ビットに0を入力させる。Wの第1ビットに1又は0を入力する直前で出力バッファレジスタシフト回路SHCが動作しシフトした後に入力される様に構成されている。

NP : 出力コントロールフラッグF/Fである。

G₁₇ : 出力コントロールフラッグF/F NPの入力ゲートである。④の時は1を入力し⑤の時は0を入力する。

G₁₈ : バッファレジスタWの出力コントロールゲートである。フラッグF/F NPがセット(1)になっている時のみ、Wの各ビットの出力を一齐に出力させる。

J : ジャッジF/Fである。

IV₁ : インバータ回路である。

IV₂ : (同上)

IV₃ : インバータ回路である。

IV₄ : (同上)

G₁₉ : ジャッジF/F_Jの入力ゲートである。
④の時に入力KN₁の状態をJに伝達する。ただしインバータIV₁を介しているのでKN₁=0の時にJ=1となる。

G₂₀ : ジャッジF/F_Jの入力ゲートである。
④の時に入力KN₂の状態をJに伝達する。ただしインバータIV₂を介しているのでKN₂=0の時にJ=1となる。

G₂₁ : ジャッジF/F_Jの入力ゲートである。
④の時に入力KF₁の状態をJに伝達する。ただしインバータIV₃を介しているのでKF₁=0の時にJ=1となる。

G₂₂ : ジャッジF/F_Jの入力ゲートである。
④の時に入力KF₂の状態をJに伝達する。ただしインバータIV₄を介しているのでKF₂=0の時にJ=1となる。

G₂₃ : ジャッジF/F_Jの入力ゲートである。
④の時に入力AKの状態をJに伝達する。

G₂₇ : ジャッジF/F_Jの入力ゲートである。
④の時キャリイF/F_Cの内容が1の時Jに1を入力する。

DC₆ : オペランドIAの解説器である。オペランドIAを解説し、メモリRAMの所望ビットの内容が1かどうかの判断を行う。

G₂₈ : メモリRAMのオペランド解説器DC₆で指定されたビット内容をジャッジF/F_Jに伝達するゲートである。④の時に動作する。RAMの指定ビットが1の時J=1となる。

V₂ : 比較回路である。アキュムレータACCの内容とオペランドIAの内容が等しいかどうかを判断し、等しい時出力1を発生する。④の時に動作する。

V₃ : 比較回路である。メモリデジットアドレスカウンタBLの内容とオペランドIAの内容が等しいかどうかを判断し、等しい時出力1を発生する。④の時動作する。

V₄ : 比較回路である。アキュムレータACC

AK=1の時、J=1となる。

G₂₄ : ジャッジF/F_Jの入力ゲートである。
④の時に入力TABの状態をJに伝達する。TAB=1の時、J=1となる。

G₂₅ : ジャッジF/F_Jのセット用ゲートである。④の時に1をJに入力する。

V₁ : 比較回路である。メモリデジットアドレスカウンタBLの内容とあらかじめ定められたデーターとを比較し、一致していれば出力1を発生する。④又は⑤が発生された時に回路が動作する。比較すべきデーターはゲートG₂₆より出力される。

G₂₆ : 比較回路V₁への比較値入力ゲートである。比較値n₁はメモリRAMの制御上の利用の高い側の特定アドレス値に対応し、n₂はメモリRAMの制御上利用の低い側の特定アドレス値に対応する。④の時はn₁を比較値にするために出力させ、⑤の時はn₂を比較値にするために出力させる。

の内容とメモリRAMの内容が等しいかどうかを判断し、等しい時、出力1を発生する。

G₂₉ : 加算第4ビットキャリイC₄のジャッジF/F_Jへの伝達ゲートである。④の時C₄をF/F_Jに伝達する。C₄=1の時、J=1となる。

FA : フラッグF/Fである。

G₃₁ : フラッグF/F_{FA}の入力ゲートである。
④の時1を出力⑤の時0を出力する。

G₃₂ : ジャッジF/F_Jの入力ゲートである。
フラッグF/F_{FA}が1の時、F/F_Jをセット(1)する。

FB : フラッグF/Fである。

G₃₃ : フラッグF/F_{FB}の入力ゲートである。
④の時1を出力し、⑤の時0を出力する。

G₃₄ : ジャッジF/F_Jの入力ゲートである。
フラッグF/F_{FB}の内容をF/F_Jに伝達する。④の時動作する。したがってFB=1の時、J=1となる。

- G 35 : ジャッジ F/F J の入力ゲートである。入力 β の内容を伝達するもので④によって動作する。 $\beta = 1$ の時 J = 1 となる。
- G 36 : アキュムレータ A C C の入力ゲートである。④の時は加算器 A D₄ の出力を伝達し、⑤の時はインバータ I V₅ にてアキュムレータの内容を反転し伝達する。⑥の時はメモリ R A M の内容を伝達し、⑦の時はオペランド I A の内容を伝達する。⑧の時は入力 $k_1 \sim k_4$ の 4 ビットの内容を伝達する。⑨の時はスタックレジスタ S A の内容を伝達する。
- I V₅ : インバータ回路である。
- S A : スタックレジスタである。出力がシステム外に導出されている。
- S X : スタックレジスタである。出力がシステム外に導出されている。
- G 37 : スタックレジスタ S A の入力ゲートである。④の時、アキュムレータ A C C の内容を伝達する。
- F E = 1 の時、J = 1 となる。
- G 43 : フラッグ F/F F E の入力ゲートである。④の時 1 を出力、⑤の時、0 を出力する。
- G 44 : ジャッジ F/F J の入力ゲートである。入力 β の内容を伝達し④によって動作する。 $\alpha = 1$ の時、J = 1 となる。
- G 45 : アキュムレータ A C C の内容を D₁/0 端子に伝達するゲートである。④によって動作する。
- G 46 : オペランド I A, I B を表示制御、キー入力制御用フラッグ N₁, N₂ に導入するゲートである。④によって動作する。
- G 47 : メモリ R A M 内の所定の数ビットの内容を伝達するゲートである。キー入力制御用フラッグ N₂ の状態により動作する。
- E O : メモリファイルアドレスカウンタ B M の内容とオペランド I A との Ex-OR をとる論理回路である。
- S B : メモリデジッアドレスカウンタ B L の内容より 1 減ずる回路である。④によ
- G 38 : スタックレジスタ S X の入力ゲートである。④の時、テンポラリレジスタ X の内容を伝達する。
- S P : プログラムスタックレジスタである。
- G 39 : プログラムスタックレジスタ S P の入力ゲートである。④の時、プログラムカウンタ P L の内容に加算器 A D₃ にて 1 を加えたものをプログラムスタックレジスタに導入する。
- F D : フラッグ F/F F D である。
- F E : フラッグ F/F F E である。
- G 40 : ジャッジ F/F J の入力ゲートである。フラッグ F/F F D の内容 F/F J に伝達する。④の時、動作する。したがって F D = 1 の時、J = 1 となる。
- G 41 : フラッグ F/F F D の入力ゲートである。④の時 1 を出力、⑤の時 0 を出力する。
- G 42 : ジャッジ F/F J の入力ゲートである。フラッグ F/F F E の内容を F/F J に伝達する。④の時、動作する。したがって
- り動作する。
- X B : 一時記憶用メモリデジッアドレスカウンタである。④により G₂ 出力は X B に記憶し、また X B の内容を D C₁ へ出力する。
- Y : 一時記憶用メモリデジッアドレスカウンタである。④により G₂ 出力は Y に記憶し、また Y の内容を D C₁ へ出力する。
- S : 一時記憶用メモリデジッアドレスカウンタである。④により G₂ 出力は S に記憶し、また S の内容を D C₁ へ出力する。
- R W : 外部メモリに対する書き込み読出し信号を発生するための信号発生回路である。④⑤によって動作する。
- P S C : 電源制御回路である。④の発生によりシステム電源 V D D を出力 (ON) する。
- Z I : メモリファイルアドレスを 0 に選択する回路である。④によって動作する。

次に、本発明実施例の説明に用いているインストラクションの記号とそのインストラクションにより指定される制御命令を表1に示す。なおインストラクションコード欄にインストラクションのワード構成を略記する。

№	インストラクションコード	インストラクション記号	制御命令
1	I ₀	SKIP	㉔
2	I ₀	AD	㉕㉖
8	I ₀	ADC	㉕㉖㉗①
4	I ₀	ADCSK	㉕㉖㉗㉘①
5	I ₀ , I _A	ADI	㉕㉖㉗
6	I ₀ , I _A	DC	㉕㉖㉗
7	I ₀	SC	㉕
8	I ₀	RC	㉕
9	I ₀ , I _A	SM	㉕
10	I ₀ , I _A	RM	㉕
11	I ₀	COMA	㉕
12	I ₀ , I _A	LDI	㉕

表1 その1

№	インストラクションコード	インストラクション記号	制御命令
13	I ₀ , I _A	L	㉕㉖
14	I ₀ , I _A	LI	㉕㉖㉗㉘㉙
15	I ₀ , I _A	LD	㉕㉖㉗㉘㉙
16	I ₀ , I _A	X	㉕㉖㉗
17	I ₀ , I _A	XI	㉕㉖㉗㉘㉙㉚
18	I ₀ , I _A	XD	㉕㉖㉗㉘㉙㉚
19	I ₀ , I _A	LBLI	㉕
20	I ₀ , I _A , I _B	LB	㉕㉖
21	I ₀ , I _A	ABLI	㉕㉖㉗
22	I ₀ , I _A	ABMI	㉕㉖
23	I ₀ , I _A	T	㉕
24	I ₀	SKC	㉕
25	I ₀ , I _A	SKM	㉕
26	I ₀ , I _A	SKBI	㉕
27	I ₀ , I _A	SKAI	㉕
28	I ₀	SKAM	㉕
29	I ₀	SKN ₁	㉕

表1 その2

№	インストラクションコード	インストラクション記号	制御命令
30	I ₀	SKN ₂	㉕
31	I ₀	SKF ₁	㉕
32	I ₀	SKF ₂	㉕
33	I ₀	SKAK	㉕
34	I ₀	SKTAB	㉕
35	I ₀	SKFA	㉕
36	I ₀	SKFB	㉕
37	I ₀	SKFD	㉕
38	I ₀	SKFE	㉕
39	I ₀	WIS	㉕
40	I ₀	WIR	㉕
41	I ₀	NPS	㉕
42	I ₀	NPR	㉕
43	I ₀	ATF	㉕
44	I ₀	LXA	㉕
45	I ₀	XAX	㉕㉖
46	I ₀	SFA	㉕

表1 その3

№	インストラクションコード	インストラクション記号	制御命令
47	I ₀	RFA	㉕
48	I ₀	SFB	㉕
49	I ₀	RFB	㉕
50	I ₀	SFC	㉕
51	I ₀	RFC	㉕
52	I ₀	SFD	㉕
53	I ₀	RFD	㉕
54	I ₀	SFE	㉕
55	I ₀	RFE	㉕
56	I ₀	SKA	㉕
57	I ₀	SKB	㉕
58	I ₀	KTA	㉕
59	I ₀	STPO	㉕
60	I ₀	EXPO	㉕㉖
61	I ₀ , I _A	TML	㉕㉖
62	I ₀	RIT	㉕
63	I ₀ , I _A , I _B	LN1	㉕

表1 その4

Ⓐ	インストラクション コード	インストラクション 記号	制御命令
64	I ₀	READ	①⑦
65	I ₀	STOR	⑦③
66	I ₀ , I _A	EX	③④⑤⑥
67	I ₀	DECB	⑦⑤④
68	I ₀	BMTA	⑤
69	I ₀	ATBM	⑥
70	I ₀	BTA	⑦
71	I ₀	ATB	③
72	I ₀	MTB	⑦
78	I ₀	SAG	③
74	I ₀	SAX	③
75	I ₀	SAY	③③
76	I ₀	SAP	③
77	I ₀ , I _A	LDY	③④⑤⑥⑦
78	I ₀	OFF	③
79	I ₀ , I _A	LDA	③④

表1 その5

キャリイF/F Cの内容を2進加算し、加算結果をアキュムレータACCに戻すと共に、この加算結果で第4ビットキャリイC₄が発生すれば次のプログラムステップをスキップする。

ADI: アキュムレータACCの内容とオペランドI_Aを2進加算し、加算結果をアキュムレータACCに戻すと共に、この加算結果で第4ビットキャリイC₄が発生すれば、次のプログラムステップをスキップする。

DC: オペランドI_Aを1010(10進数10)に定め、ADI命令と同様に、アキュムレータACCの内容と、このオペランドI_Aを2進加算することによって実質的にアキュムレータACCの内容に10進数10を加算し、その結果をアキュムレータACCに戻す。

SC: キャリイF/F Cをセットする(Cに1を入力する)。

RC: キャリイF/F Cをリセットする(Cに0を入力する)。

SM: オペランドI_Aの内容を解読し、オペラン

Ⓐ	インストラクション コード	インストラクション 記号	制御命令
80	I ₀	ROT	③③
81	I ₀	INCB	①③④
82	I ₀ , I _A	EXCI	③①③③③④④
83	I ₀ , I _A	EXCD	③①③③③④④

表1 その6

表1に挙げたインストラクションの命令内容は次の通りである。

SKIP: 次のプログラムステップの命令を実行せず、プログラムカウンタPLのみアップさせ実質的にスキップする。

AD: アキュムレータACCの内容とメモリRAMの内容を2進加算し、加算結果をアキュムレータACCに戻す。

ADC: アキュムレータACC、メモリRAM、キャリイF/F Cの内容を2進加算し、加算結果をアキュムレータACCに戻す。

ADCSK: アキュムレータACC、メモリRAM、

Dで指定されたメモリの所望ビットをセットする(1を入力する)。

RM: オペランドI_Aの内容を解読し、オペランドで指定されたメモリの所望ビットをリセットする(0を入力する)。

COMA: アキュムレータACCの各ビットの内容を反転し、15の補数を取りアキュムレータACCに戻す。

LDI: アキュムレータACCにオペランドI_Aを導入する。

L: メモリRAMの内容をアキュムレータACCに導入すると共に、オペランドI_AをメモリアドレスカウンタB_Mに入力する。

LI: メモリRAMの内容をアキュムレータACCに導入すると共に、オペランドI_AをメモリアドレスカウンタB_Mに入力する。さらにメモリデジットアドレスカウンタB_Lをアップさせる。ただしB_Lの内容があらかじめ定めた値n₁に等しい時は次のプログラムステップをスキップする。

LD: メモリRAMの内容をアキュムレータACCに導入すると共に、オペランドIAをメモリファイルアドレスカウンタBMに入力する。さらにメモリディジットアドレスカウンタBLをダウンさせる。ただしBLの内容があらかじめ定められた値n₂に等しい時は次のプログラムステップをスキップする。

X: メモリRAMの内容とアキュムレータACCの内容を交換すると共に、オペランドIAをメモリファイルアドレスカウンタBMに入力する。

XI: メモリRAMの内容とアキュムレータACCの内容を交換すると共に、オペランドIAをメモリファイルアドレスカウンタBMに入力する。さらにメモリディジットアドレスカウンタBLをアップさせる。ただしBLの内容があらかじめ定められた値n₁に等しい時は次のプログラムステップをスキップする。

XD: メモリRAMの内容とアキュムレータACCの内容を交換すると共に、オペランドIAをメモリファイルアドレスカウンタBMに入力する。

SKC: キャリイF/FCが1ならば次のプログラムステップをスキップする。

SKM: オペランドIAの内容を解読し、オペランドで指定されたメモリの所望ビットが1であれば、次のプログラムステップをスキップする。

SKBI: メモリディジットアドレスカウンタBLの内容と、オペランドIAを比較し、等しい時には次のプログラムステップをスキップする。

SKAI: アキュムレータACCの内容と、オペランドIAを比較し、等しい時には次のプログラムステップをスキップする。

SKAM: アキュムレータACCの内容と、メモリRAMの内容を比較し、等しい時には次のプログラムステップをスキップする。

SKN₁: KN₁入力が0の時、次のプログラムステップをスキップする。

SKN₂: KN₂入力が0の時、次のプログラムステップをスキップする。

SKF₁: KF₁入力が0の時、次のプログラムステップをスキップする。

さらにメモリディジットアドレスカウンタBLをダウンさせる。ただしBLの内容があらかじめ定められた値n₂に等しい時は次のプログラムステップをスキップする。

LBLI: オペランドIAをメモリディジットアドレスカウンタBLに入力する。

LB: オペランドIAをメモリファイルアドレスカウンタBMに入力すると共に、オペランドIBをメモリディジットアドレスカウンタBLに入力する。

ABLI: メモリディジットアドレスカウンタBLの内容とオペランドIAを2進加算し、加算結果をBLに戻す。ただしBLの内容があらかじめ定められた値n₁に等しい時は、次のプログラムステップをスキップする。

ABMI: メモリファイルアドレスカウンタBMの内容とオペランドIAを2進加算し、加算結果をBMに戻す。

T: オペランドIAをプログラムステップカウンタPLに入力する。

SKF₂: KF₂入力が0の時、次のプログラムステップをスキップする。

SKAK: AK入力が1の時、次のプログラムステップをスキップする。

SKTAB: TAB入力が1の時、次のプログラムステップをスキップする。

SKFA: フラッグF/FF_Aが1の時、次のプログラムステップをスキップする。

SKFB: フラッグF/FF_Bが1の時、次のプログラムステップをスキップする。

SKFD: フラッグF/FF_Dが1の時、次のプログラムステップをスキップする。

SKFE: フラッグF/FF_Eが1の時、次のプログラムステップをスキップする。

WIS: バッファレジスタWの内容を1ビット右シフトすると共に、第1ビット(最上位ビット)に1を入力する。

WIR: バッファレジスタWの内容を1ビット右シフトすると共に、第1ビット(最上位ビット)に0を入力する。

- NPS: バッファレジスタWの出力コントロールF/FNPをセットする(1を入力する)。
- NPR: バッファレジスタWの出力コントロールF/FNPをリセットする(0を入力する)。
- ATF: アキュムレータACCの内容をバッファレジスタFに転送する。
- LXA: アキュムレータACCの内容をテンポラリーレジスタXに転送する。
- XAX: アキュムレータACCの内容とテンポラリーレジスタXの内容を交換する。
- SFA: フラッグF/FFAをセットする(1を入力する)。
- RFA: フラッグF/FFAをリセットする(0を入力する)。
- SFB: フラッグF/FFBをセットする(1を入力する)。
- RFB: フラッグF/FFBをリセットする(0を入力する)。
- SFC: 入力テスト用フラッグF/FFCをセットする(1を入力する)。
- RFC: 入力テスト用フラッグF/FFCをリセットする(0を入力する)。
- SFD: 入力テスト用フラッグF/FFDをセットする(1を入力する)。
- RFD: 入力テスト用フラッグF/FFDをリセットする(0を入力する)。
- SFE: 入力テスト用フラッグF/FFEをセットする(1を入力する)。
- RFE: 入力テスト用フラッグF/FFEをリセットする(0を入力する)。
- SKA: 入力 α が1の時、次のプログラムステップをスキップする。
- SKB: 入力 β が1の時、次のプログラムステップをスキップする。
- KTA: 入力 $k_1 \sim k_4$ の内容をアキュムレータACCに導入する。
- STPO: アキュムレータACCの内容をスタックレジスタSAに、テンポラリーレジスタXの内容をスタックレジスタSXに転送する。
- EXPO: アキュムレータACCの内容とスタックレジスタSAの内容を交換し、テンポラリーレジスタXの内容とスタックレジスタSXの内容を交換する。
- TML: プログラムカウンタPLの内容に1を加えたものをプログラムスタックレジスタSPに転送する。さらにオペランドIAをプログラムカウンタに導入する。
- RIT: プログラムスタックレジスタSPの内容をプログラムカウンタPLに転送する。
- LNI: 表示制御、キー入力制御用フラッグN₁ N₂にオペランドIA, IBを導入する。
- READ: 外部よりDI/Oに入力されるデータをアキュムレータACCに導入する。
- STOR: アキュムレータACCの内容をDI/Oに出力する。
- EX: メモリRAMの内容とアキュムレータACCの内容を交換すると共にオペランドIAとメモリアドレスカウンタBMの内容とのEX-ORをBMに入れる。
- DECB: メモリディジットアドレスカウンタBLの内容をカウントダウンする。ただしBLの内容があらかじめ定められた値n₂に等しい時は次の命令をスキップする。
- BMTA: メモリアドレスカウンタBMの内容をアキュムレータACCに導入する。
- ATBM: アキュムレータACCの内容をメモリアドレスカウンタBMに導入する。
- BTA: メモリディジットアドレスカウンタBL, XB, Y, Sのうち指定されたカウンタの内容をアキュムレータACCに導入する。
- ATB: アキュムレータACCの内容をメモリディジットアドレスカウンタBL, XB, Y, Sのうち指定されたカウンタに導入する。
- MTB: メモリRAMの内容をメモリディジットアドレスカウンタBL, XB, Y, Sのうち指定されたカウンタに導入する。
- SAG: 次のステップで指定するメモリのファイルアドレスを0000とする。
- SAX: 次のステップで指定するメモリのディジットアドレスはXBの内容とする。

SAY: 次のステップで指定するメモリのディジットアドレスはYの内容とし、ファイルアドレスを0000とする。

SAP: 次のステップで指定するメモリのディジットアドレスはSの内容とする。

LDY: メモリRAMの内容をアキュムレータACCに導入、メモリファイルアドレスカウンタ B_M とオペランド I_A とのEx-ORを B_M に入れ B_L, X, Y, S のうち指定されたカウンタを一つカウントアップする。ただし指定されたカウンタの内容があらかじめ定められた値 n_1 に等しい時は次のプログラムステップをスキップする。また次のステップで指定するメモリのディジットアドレスはYの内容とする。

OFF: システム電源 V_{DD} をOFFする。出力バッファ内蔵RAM電源コントロール部等には電源の供給を続ける。

LDA: メモリの内容をアキュムレータに導入すると共にオペランド I_A とメモリファイルアドレスカウンタ B_M の内容とのEx-ORを B_M に

入れる。

ROT: アキュムレータの内容をCF/Fと連結して右シフトする。

INCB: メモリディジットアドレスカウンタ B_L の内容をカウントアップする。ただし B_L の内容があらかじめ定められた値 n_1 に等しい時は次の命令をスキップする。

EXCI: アキュムレータの内容とメモリの内容を交換すると共にメモリファイルアドレスカウンタ B_M とオペランド I_A とのEx-ORを B_M に入れディジットアドレスカウンタ B_L をカウントアップする。ただし $B_L=n_1$ の時次の命令をスキップする。

EXCD: アキュムレータの内容とメモリの内容を交換すると共にメモリファイルアドレスカウンタ B_M とオペランド I_A とのEx-ORを B_M に入れディジットアドレスカウンタ B_L をカウントダウンする。ただし $B_L=n_2$ の時次の命令をスキップする。

次に、本発明のマイクロプロセッサによる基本的な情報処理の手順を説明する。

1) 同じ数値Nをメモリの所望領域に導入する手段として下記のType1~Type3のいずれかを用いることができる。

(Type 1)

P ₁	LB	m _A	n _B
P ₂	LDI		N
P ₃	XD	m _A	
P ₄	T		P ₂

表2

表2において、

P₁は、メモリの処理すべき第1番目のディジットを、ファイルアドレス m_A とディジットアドレス n_B で指定する。(第26図参照)

P₂は、ACCに数値Nを導入する。

P₃は、メモリとACCの内容を交換することによって数値Nをメモリの指定された領域に導入す

る。メモリのファイルアドレスは変わらないので m_A を指定し、ディジットアドレスは次の導入すべきディジットを決めるためにダウンされる。導入すべき最終ディジット n_A の値をあらかじめ n_2 として決めておくことによって、数値Nを所望全領域に導入し終えた状態で $B_L=n_2$ となるため、次のP₄をスキップしてType1の処理を終える。P₄は、プログラムアドレスをP₂に指定して、 $B_L=V$ になるまで、LDIとXDの処理を繰り返す。

(Type 2)

P ₁	LB	m _B	n _C
P ₂	LDI		N
P ₃	XD		

表3

表3において、

P₁は、メモリの処理すべきディジットを、ファイルアドレス m_B と、ディジットアドレス n_C で

指定する。

P₂ は、ACC に数値 N を導入する。

P₃ は、メモリと ACC の内容を交換することによって、数値 N をメモリの指定された領域に導入する。こうして Type 2 の処理を終える。X_D のオペランド部分は続く処理に必要なもので、本処理には関係ない。

(Type 3)

P ₁	LB	mc	nd
P ₂	LDI	N	
P ₃	XD	mc	
P ₄	SKBI	n _A	
P ₅	T	P ₂	

表 4

表 4 において、

P₁ は、メモリの処理すべき第 1 番目のディジットをファイルアドレス mc と、ディジットアドレス nd で指定する。

できる。

(Type 1)

P ₁	LB	mA	n _B
P ₂	LDI	N ₁	
P ₃	XI	mA	
P ₄	LDI	N ₂	
P ₅	XI	mA	
P ₆	LDI	N ₃	
P ₇	XI	mA	
P ₈	LDI	N ₄	
P ₉	XI	mA	

表 5

表 5 は 4 桁の数値 N₄N₃N₂N₁ をメモリに導入する例であるが任意桁の導入も同じ考え方である。

P₁ は、メモリの処理すべき第 1 番目のディジットを、ファイルアドレス mA とディジットアドレス n_B で指定する。(第 27 図参照)

P₂ は、ACC に第 1 の定数 N₁ を導入する。

P₂ は、ACC に数値 N を導入する。

P₃ は、メモリと ACC の内容を交換することによって数値 N をメモリの指定された領域に導入する。メモリのファイルアドレスは変わらないので mc を指定し、ディジットアドレスは次の導入すべきディジットを決めるためにダウンされる。

P₄ は、P₃ で処理したディジットが最終ディジット n_B であったかどうかのチェックで、n_B であった時、ディジットアドレスはダウンして n_A になっているため、SKBI 命令のオペランド部分を n_A にしておくことによって最終ディジットに数値 N を導入して、P₄ に進んだ際、条件が満足し、次のアドレス P₅ をスキップして Type 3 を終了する。条件が満足しない時は P₅ に進む。

P₅ は、プログラムアドレスを P₂ に指定し BL = n_A になるまで P₂~P₄ の処理をくりかえす。

2) あらかじめ定められた複数の異なる数値をメモリの所望領域に導入する手段として下記の Type 1~Type 2 のいずれかを用いることが

P₃ は、メモリと ACC の内容を交換することによって数値 N₁ をメモリの指定された領域に導入する。メモリのファイルアドレスは変わらないので mA を指定し、ディジットアドレスは次の導入すべきディジットを決めるためにアップする。

P₄ は、ACC に第 2 の定数 N₂ を導入する。

P₅ は、P₃ の処理でメモリは第 2 番目のディジットに指定されているため、メモリと ACC の内容交換によって、第 2 の定数 N₂ がメモリの第 2 番目のディジットに導入される。

P₆ 以下は上記と同様である。

(Type 2) 0~15 のうちの任意の数値をあらかじめ定めたレジスタに導入する。

P ₁	LDI	N	
P ₂	LXA		

表 6

表 6 において、

P₁ は、ACC に数値 N を導入する。

P₂ は、ACC に入っている数値 N をテンポラリレジスタ X に導入する。

3) メモリの所望領域の内容をメモリの他の所望領域に転送する手段として下記の Type 1 ~ Type 3 のいずれかを用いることができる。

(Type 1)

X → Y

P ₁	LB	m _A	n _E
P ₂	L	m _B	
P ₃	XI	m _A	
P ₄	T	P ₂	

表 7

表 7 において、

P₁ は、処理すべき第 1 のメモリのファイルアドレスを m_A で指定し、処理すべき第 1 のディジットアドレスを n_E で指定する。(第 28 図参照)
P₂ は、第 1 のメモリの所望ディジットの内容を

ACC に導入すると共に、P₃ での転送処理に備えて、転送先の第 2 のメモリのファイルアドレスを m_B で指定する。

P₃ は、ACC に導入した第 1 のメモリの内容を P₂ で指定した第 2 のメモリの同一ディジットの内容と交換して、実質的に第 1 メモリの内容を第 2 メモリに転送する。同時にくり返してこの処理をするためにもとの第 1 のメモリのファイルアドレスを m_A で指定しておく。転送すべき最終ディジット n_A の値をあらかじめ n₁ として決めておくことによって、第 1 のメモリ内容を全て第 2 のメモリに転送し終えた状態で B_L=n₁ となるため、次の P₄ をスキップして Type 1 の処理を終える。B_L=V になるまではディジットアドレスを順次アップして P₄ を介して P₂ に戻るファイルアドレスを m_A にしておき、第 1 メモリを指定する。

P₄ は、プログラムアドレスを P₂ に指定して、B_L=n₁ になるまで P₂ と P₃ の命令をくり返し、1 ディジット毎、転送処理を進めてゆく。

(Type 2) X_n → Y_m

P ₁	LB	m _B	n _C
P ₂	L	m _C	
P ₃	LBI	n _D	
P ₄	X		

表 8

表 8 において、

P₁ は、処理すべきメモリの領域をファイルアドレス m_A とディジットアドレス n_C で指定する。(第 29 図参照)

P₂ は、P₁ で指定したメモリ領域の内容を ACC に導入すると共に、P₄ での転送処理に備えて転送先のメモリのファイルアドレスを m_C で指定する。

P₃ は、転送先のメモリのディジットアドレスを指定する。P₂ と P₃ の処理で転送先のメモリの領域を決定する。

P₄ は、ACC の内容を P₂, P₃ で指定されたメ

モリの領域と交換し、実質的に転送する。X のオペランドは本処理には直接関係しない。

(Type 3)

P ₁	LB	m _B	n _C
P ₂	L		
P ₃	LXA		

表 9

表 9 において、

P₁ は、処理すべきメモリの領域をファイルアドレス m_A とディジットアドレス n_C で指定する。(第 30 図参照)

P₂ は、P₁ で指定したメモリ領域の内容を ACC に導入する。

P₃ は、ACC に導入されたメモリの内容をレジスタ X に導入し、所望の Type 3 の転送処理を実行する。

4) メモリの所望領域の内容とメモリの他の所望

領域の内容とを交換する手段として下記の
Type 1~Type 4のいずれかを用いることが
できる。

(Type 1) $X = Y$

P ₁	LB	m _A	n _E
P ₂	L	m _B	
P ₃	X	m _A	
P ₄	XI	m _A	
P ₅	T	P ₂	

表 10

表 10 において、

P₁ は、処理すべき第 1 のメモリのファイルアドレスを m_A で指定し、処理すべき第 1 のディジットアドレスを n_E で指定する。(第 31 図参照)
P₂ は、第 1 のメモリの所望ディジットの内容を ACC に導入すると共に、P₃ での第 2 のメモリとの交換処理に備えて、第 2 のメモリのファイルアドレスを m_B で指定する。

P₅ は、プログラムアドレスを P₂ に指定し、
BL=n₁ になるまで P₂~P₄ の命令をくり返し、
1 ディジット毎、交換処理を進めてゆく。

(Type 2) $X_n = Y_m$

P ₁	LB	m _B	n _C
P ₂	L	m _C	
P ₃	LBLI	n _D	
P ₄	X	m _B	
P ₅	LBLI	n _C	
P ₆	X		

表 11

表 11 において、

P₁ は、処理すべき第 1 のメモリのファイルアドレスを m_A で指定し、処理すべきディジットアドレスを n_C で指定する。(第 32 図参照)
P₂ は、第 1 のメモリの所望ディジットの内容を ACC に導入すると共に、第 2 メモリのファイルアドレス m_C を指定し、内容交換に備える。

P₃ は、ACC に入っている第 1 メモリの所望ディジットの内容と、P₂ で指定された第 2 のメモリの同一ディジットの内容を交換すると共に、この処理で ACC に転送された第 2 のメモリの内容を第 1 のメモリに導入するために第 1 メモリのファイルアドレスを m_A で指定しておく。

P₄ は、ACC に導入された第 2 メモリの内容と、同一ディジットの第 1 メモリの内容とを交換し、第 2 メモリの内容を第 1 メモリに転送する。P₂~P₄ の処理にてメモリ所望ディジット間の内容交換を行う。第 1 メモリの指定はファイルアドレス m_A の指定にて継続させ、ディジットアドレスをアップさせ、次のディジットアドレスを指定し、交換を各ディジットに対して順次実行してゆく。なお交換すべき最終ディジット n_A の値をあらかじめ n₁ として決めておくことによって、第 1 のメモリと第 2 のメモリの内容を全ディジットにわたって交換し終えた状態で、BL=n₁ となるため、次の P₅ をスキップして、Type 1 の処理を終える。

P₃ は、転送先の第 2 メモリのディジットアドレス n_D を指定し、交換先のメモリアドレスを決定する。

P₄ は、ACC に入っている第 1 メモリの内容と第 2 メモリの内容を交換する。この時 ACC に転送される第 2 メモリの内容を第 1 メモリに転送させるため再び第 1 メモリのファイルアドレスを m_B で指定する。

P₅ は、第 1 メモリのディジットアドレス n_C を指定し、転送先の第 1 メモリアドレスを決定する。

P₆ は、ACC に入っている第 2 メモリの内容と第 1 メモリの内容を交換し、第 1 メモリと第 2 メモリの内容の交換を実行する。

(Type 3) $X_n = Y_n$

P ₁	LB	m _B	n _C
P ₂	L	m _C	
P ₃	X	m _B	
P ₄	X		

表 12

表12において、

P₁は、処理すべき第1メモリのファイルアドレスをm_Aで指定し、処理すべきディジットアドレスをn_Cで指定する。(第33図参照)

P₂は、第1のメモリ内容をACCに導入すると共に、交換先に第2メモリのファイルアドレスm_Cで指定する。

P₃は、ACCの第1メモリの内容と、P₂で指定された第2メモリの内容を交換し、第1メモリ内容を第2メモリに導入する。P₄での処理に備え、再び第1メモリをファイルアドレスm_Bで指定しておく。

P₄は、ACCに導入された第2メモリの内容と第1メモリの内容を交換することによって第1メモリと第2メモリの内容交換を実行する。

表13において、

P₁は、処理すべきメモリの領域をファイルアドレスm_Bとディジットアドレスn_Cを指定する。(第34図参照)

値Nを2進加算又は減算する手段として下記のType1~Type9のいずれかを用いることができる。

(Type 1) $M_1 + N \rightarrow M_1$

P ₁	LB	m _B	n _C
P ₂	L	m _B	
P ₃	ADI		N
P ₄	X		

表14

表14において、

P₁は、メモリの処理すべき領域をファイルアドレスm_Bとディジットアドレスn_Cで指定する。

(第35図参照)

P₂は、P₁で指定されたメモリの内容をACCに導入する。メモリファイルアドレスの指定は後に再び同じメモリに戻すためm_Bを指定しておく。

P₃は、オペランドで加算すべき数値Nを指定し、ACCに導入されたメモリの内容と数値Nを加算

(Type 4)

P ₁	LB	m _B	n _C
P ₂	L	m _B	
P ₃	XAX		
P ₄	X		

表13

P₂は、P₁で指定されたメモリの内容をACCに導入する。レジスタXの内容との交換に備え、ファイルアドレスm_Bを維持しておく。

P₃は、ACCに入っているメモリの内容とレジスタXの内容を交換し、レジスタXにメモリの内容を転送する。

P₄は、ACCに入っているレジスタXの内容をメモリと交換することによって、レジスタXの内容を実質的にメモリに転送し、Type4を実行させる。

5) メモリの所望領域にあらかじめ定められた数

し、その結果をACCに求める。

P₄は、ACCに求められた和をP₂で指定したもとのメモリの内容とを交換し、Type1を実行する。

(Type 2) $X + N \rightarrow X$

P ₁	XAX		
P ₂	ADI		N
P ₃	XAX		

表15

表15において、

P₁は、レジスタXの内容とACCの内容を交換する。

P₂は、オペランドで加算すべき数値Nを指定し、ACCに導入されたレジスタXの内容と数値Nを加算し、その結果をACCに求める。

P₃は、ACCに求められた和とレジスタXの内容を交換することによって実質的に $X + N \rightarrow N$ なるType2を実行する。

(Type 3) $M_1 + N \rightarrow M_2$

P ₁	LB	m _B	n _C
P ₂	L	m _C	
P ₃	ADI	N	
P ₄	X		

表16

表16において、

P₁は、第1メモリの処理すべき領域をファイルアドレスn_Cで指定する。

P₂は、P₁で指定されたメモリの内容をACCに導入する。メモリファイルアドレスの指定は、加算結果を第2メモリに戻すため第2メモリのファイルアドレスm_Cを指定しておく。

P₃は、オペランドで加算すべき数値Nを指定し、ACCに導入されたメモリの内容を数値Nと加算し、その結果をACCに求める。

P₄は、ACCに求められた和をP₂で指定した第2メモリの内容と交換し、Type 3を実行する。

減数の16の補数と被減数を加算する処理で置換される。ボローのない状態をC=1とし、 $ACC + C + M \rightarrow ACC$ にて純2進の減算が実行される。

P₆は、P₅で求まった差を同じメモリに戻すためACCとメモリを交換する。

(Type 5) $M_1 - N \rightarrow M_2$

P ₁	LB	m _B	n _C
P ₂	SC		
P ₃	LDI	N	
P ₄	COMA		
P ₅	ADC		
P ₆	LB	m _C	n _C
P ₇	X		

表18

表18において、

P₁~P₅はType 4と同様である。

P₆は、P₅で求まった差を第2メモリに導入するため、第2メモリのファイルアドレスm_Cとデ

(Type 4) $M_1 - N \rightarrow M_1$

P ₁	LB	m _B	n _C
P ₂	SC		
P ₃	LDI	N	
P ₄	COMA		
P ₅	ADC		
P ₆	X		

表17

表17において、

P₁は、処理すべきメモリのファイルアドレスm_Bとデジットアドレスn_Cを指定する。

P₂は、減算は減数の補数を被減数に加える方式で、下位桁がないのでボローがなくF/F Cをセットしておく。

P₃は、ACCに減数Nを導入する。

P₄は、減数の15の補数をとるための処理で、補数がACCに求まる。

P₅は、減算は下位桁からのボローがなければ、

デジットアドレスn_Cを指定する。

P₇は、P₆で指定された第2メモリに、ACCに求まっている差データを交換によって転送する。

(Type 6) $X - N \rightarrow X$

P ₁	LB	m _B	n _C
P ₂	SC		
P ₃	LDI	N	
P ₄	COMA		
P ₅	X	m _B	
P ₆	XAX		
P ₇	ADC		
P ₈	XAX		

表19

表19において、

P₁は、P₅での一時待選メモリのアドレスをファイルアドレスm_Bとデジットアドレスn_Cで指定する。

P₂は、減算は減数の補数を被減数に加える方式

で、下位桁がないのでボローがなくF/F Cをセットしておく。

P₃は、ACCに減数Nを導入する。

P₄は、減数の15の補数をとるための処理で、補数がACCに求まる。

P₅は、レジスタXの内容との演算に備え、P₁で指示したメモリにACCの内容を導入する。

P₆は、レジスタXの内容をACCとの交換にて転送する、この処理を終えるとメモリには減数の15の補数、ACCにはXの内容が入っている。

P₇は、ACC+M+CはX-Nに相当する処理で2進の実質的な減算結果がACCに求まる。

P₈は、ACCの内容とXの内容を交換し、X-Nの値をXに転送し、Type 6の処理を終える。

表20において、

P₁は、処理すべきメモリのファイルアドレスm_Bとデジットアドレスn_Cを指定する。

P₂は、1デジット分の減算であり、減数の補数を被減数に加える方式なのでF/F Cをセット

しておく。

(Type 7) $N - M_1 \rightarrow M_1$

P ₁	LB	m _B	n _C
P ₂	SC		
P ₃	LDI	N	
P ₄	X	m _B	
P ₅	COMA		
P ₆	ADC		
P ₇	X		

表20

P₃は、ACCに被減数を導入する。

P₄は、メモリの内容(減数)とACCを交換し、又P₇の処理に備え、メモリファイルアドレスはm_Bのままとしておく。

P₅は、ACCの減数の15の補数をとるための処理で、補数がACCに求まる。

P₆は、減算は下位桁からのボローがなければ、減数の16の補数と被減数を加算する処理で置換

される。ボローのない状態をC=1とし、ACC+C+Mにて、実質的にN-Mを行い、ACCにその差を求める。

P₇は、P₄でメモリファイルアドレスはそのままm_BになっているのでACCの差がもとのメモリに入り、Type 7を実行し終える。

(Type 8) $N - M_1 \rightarrow M_2$

P ₁	LB	m _B	n _C
P ₂	L	m _C	
P ₃	COMA		
P ₄	ADI	N+1	
P ₅	X		

表21

表21において、

P₁は、処理すべきメモリのファイルアドレスm_Bとデジットアドレスn_Cを指定する。

P₂は、P₁で指定した減数に相当する内容をACCに導入する。P₅の処理に備え第2メモリ

のファイルアドレスm_Cを指定しておく。

P₃は、ACCの減数15の補数をとるための処理で、補数がACCに求まる。

P₄は、オペランドの内容は被減数に1を加えたものに設定しておく。これはこの減算が1デジット分のものであり、減数の補数と被減数を加算する処理で置換される。ボローのない状態での一般的な補数加算はType 7の如く $\overline{ACC} + C + M$

であり、C=1として処理される。ADI命令ではCがないのであらかじめ $\overline{ACC} + 1$ を行って処理する。これによってN-MのType 8の演算結果がACCに求まる。

P₅は、P₄で求められた差データをP₂で指定した第2メモリに転送する。

表22において、

P₁は、(M+1の時)ACCに2進数0001 (=1)を導入する。

P_{1'}は、(M-1の時)ACCに2進数1111 (=15)を導入する。

(Type 9) $M \pm 1 \rightarrow M$

P ₁	LDI	I
P ₁ '	LDI	F
P ₂	LB	mB nC
P ₃	AD	
P ₄	X	

表 2 2

P₂ は、処理すべきメモリのファイルアドレス mB とデジジットアドレス nC を指定する。

P₃ は、P₂ で指定されたメモリの内容と P₁ 又は P₁' で ACC の導入された内容を加算し ACC に和を導入する。P₁ の場合は ACC+1 になり、P₁' の場合は実質的に ACC-1 になる。

P₄ は、ACC に求められた結果をもとのメモリに転送し、Type 9 を終える。

- 6) メモリの所望領域の内容に他の領域の内容を 10 進加算又は減算する手段として下記

ておく。

P₃ は、第 1 メモリの所望デジジットの内容を ACC に導入すると共に、P₄ での第 2 メモリの内容との加算に備えて、ファイルアドレスを第 2 メモリの mB に指定しておく。

P₄ は、ACC に導入した第 1 メモリの所望デジジットの内容に 6 を加え、P₅ での加算時の次桁への 10 進桁上の有無判断のために用いる。

P₅ は、P₄ で第 1 メモリに 6 補正をしたものが ACC に求められていて、この ACC の内容と P₃ で指定した第 2 メモリの同一デジジットの内容とを純 2 進加算し、再び ACC に導入する。この純 2 進加算の第 4 ビット目の加算で桁上が出た場合、P₆ をスキップして P₇ へ進む。第 4 ビット目の加算で桁上ができることは、10 進桁上があったことを意味する。

P₆ は、P₅ の加算で 10 進桁上が出なかった時、P₄ で加算した 6 をこのステップで減じてもとの値に戻す。10 の加算は 6 の減算と同じである。

P₇ は、ACC に求まっている 10 進の 1 桁分の

Type 1~Type 2 のいずれかを用いることができる。

(Type 1) $X + W \rightarrow X$

P ₁	LB	mA nE
P ₂	RC	
P ₃	L	mB
P ₄	ADI	6
P ₅	ADCSK	
P ₆	DC	
P ₇	XI	mA
P ₈	T	P ₃

表 2 3

表 2 8 において、

P₁ は、処理すべき第 1 メモリの第 1 デジジットをファイルアドレス mA とデジジットアドレス nE で指定する。

P₂ は、第 1 デジジットの加算の際、下位桁からの桁上処理はないため桁上 F/F C をリセットし

和を第 2 メモリに交換によって転送すると共に、次桁の加算に備えデジジットアドレスをアップさせ、さらに第 1 メモリをファイルアドレス mA で指定しておく。加算すべき最終デジジットをあらかじめ n₁ として決めておくことによって、第 1 メモリと第 2 メモリの全デジジットの加算を終えた状態で B_L=n₁ となるため、次の P₈ をスキップして Type 1 の処理を終える。

P₈ は、プログラムアドレス P₃ を指定して、B_L=n₁ になるまで P₃~P₇ の命令をくり返し、1 デジジット毎、10 進加算を進めてゆく。

表 2 4 において、

P₁ は、処理すべき第 1 のメモリの第 1 デジジットをファイルアドレス mA とデジジットアドレス nE で指定する。

P₂ は、減算は減数の補数を被減数に加える方式で、第 1 デジジットの減算では下位桁からのボローの処理がないため、F/F C をセットしておく。

P₃ は、第 1 メモリの所望デジジットの減数とな

(Type 2) $X - W - X$

P ₁	LB	m _A	n _E
P ₂	SC		
P ₃	L	m _B	
P ₄	COMA		
P ₅	ADCSK		
P ₆	DC		
P ₇	XI	m _A	
P ₈	T		P ₃

表 24

る内容をACCに導入すると共に、P₅、P₇での第2メモリとの処理に備えて、第2メモリファイルアドレスm_Bを指定しておく。

P₄は、減数の15の補数をとるための処理である。15の補数がACCに求められる。

P₅は、減算は下位桁からのボローがなければ、減数の16の補数と被減数を加算する処理で置換され、下位桁からのボローがあれば減数の15の

次のP₈をスキップしてType 2の処理を終える。

P₈は、プログラムアドレスP₃を指定して、B_L=n₁になるまでP₃~P₇の命令をくり返し、1ディジット毎、10進減算を進めてゆく。

7) 所望領域のメモリの内容を1ディジットシフトする手段として下記のType 1~Type 2のいずれかを用いることができる。

(Type 1) 右シフト

P ₁	LB	m _A	n _A
P ₂	LDI		0
P ₃	XD	m _A	
P ₄	T		P ₃

表 25

表 25 において、

P₁は、処理すべきメモリのファイルアドレスm_Aと最上位ディジットアドレスn_Aを指定する。

P₂は、0をACCに導入し、右シフトした時、

補数と被減数との加算で置換される。ボローのない状態をC=1とし、 $\overline{ACC}+C+M \rightarrow ACC$ にて純2進の減算が実行される。このADCSKの命令実行結果キャリイが出ることは減算にてボローが出なかったことを意味するので、P₆をスキップしてP₇へ進む。なおここでの加算はP₃で指定した第2メモリとの間で行われるので実質的に(第2メモリ-第1メモリ)となる。

P₆は、P₅のADCSK命令でキャリイが出なかった場合、結果は16進数で求まっているため6を減じる(10を加えるのと同様)ことによって10進数に戻す。

P₇は、ACCに求まった第2メモリと第1メモリの差を第2メモリの内容との交換によって転送する。次の減算に備え、ディジットアドレスをアップさせ、さらに第1メモリをファイルアドレスm_Aで指定しておく。減算すべき最終ディジットをあらかじめn₁として決めておくことによって、第2メモリと第1メモリの減算を全ディジットにわたって終えた状態でB_L=n₁となるため、

最上位ディジットに0を入れる為の準備をする。

P₃は、ACCとメモリの内容を交換すると共にディジットアドレスをダウンさせ、1ディジット下位を指定する。メモリファイルアドレスはm_Aで変えない。次のP₄を介して再びP₃に戻るのでXDのくり返しを意味する。P₂でACCに入れた0は最初のACC=Mにてメモリの最上位ディジットに入り、もとの最上位ディジットにあった内容はACCに入る。P₃でディジットアドレスがダウンされP₄を介してP₃に戻ってXDを実行した時、最上位より1ディジット下位が指定されているので、ACCに入っているもとの最上位ディジットの内容が1ディジット下位に転送される。この時ACCには最上位より1ディジット下位の内容が転送されている。最上位ディジットをあらかじめn₂と決めておくことによって、上記転送を最下位ディジットまでくり返すと、B_L=n₂が満足し、P₄をスキップして終える。すなわち1ディジット毎の内容が下位ディジットに転送され、Type 1を実行する。

P₄ は、B_L=VになるまでP₃のX Dをくり返すためP₃に戻る。

(Type 2) 左シフト

P ₁	LB	m _A	0E
P ₂	LDI		0
P ₃	XI	m _A	
P ₄	T		P ₃

表 26

表 26 において、

P₁ は、処理すべきメモリのファイルアドレスm_Aと最下位ディジットn_Eを指定する。

P₂ は、0をACCに導入し、左シフトした時、最下位ディジットに0を入れる準備をする。

P₃ は、ACCとメモリの内容を交換すると共に、ディジットアドレスをアップさせ、1ディジット上位を指定する。メモリファイルアドレスはm_Aで変えない。次のP₄を介して再びP₃に戻るの

ることができる。

(Type 1)

P ₁	LB	m _B	n _C
P ₂	SM		N

表 27

表 27 において、

P₁ は、メモリの処理すべき領域のディジットをファイルアドレスm_Bとディジットアドレスn_Cで指定する。

P₂ は、P₁で指定されたメモリのディジットの中の所望ビットNに対して1を導入し、Type 1を実行する。

(Type 2)

P ₁	LB	m _B	n _C
P ₂	RM		N

表 28

れた0は最初のACC=Mでメモリの最下位ディジットに入り、もとの最下位ディジットにあった内容はACCに入る。P₃でディジットアドレスがアップされP₄を介してP₃に戻ってXIを実行した時、最下位より1ディジット上位が指定されているので、ACCに入っているもとの最下位ディジットの内容が1ディジット上位に転送される。この時ACCには最下位より1ディジット上位の内容が転送されている。最上位ディジットをあらかじめn₁と決めておくことによって、上記転送を最上位ディジットまでくり返すとB_L=n₁が満足し、P₄をスキップして終える。すなわち1ディジット毎、内容が上位ディジットに転送され、Type 2を実行する。

P₄ は、B_L=VになるまでP₃のXIをくり返すためにP₃に戻る。

8) メモリの所望領域の1ビットコンディショナルF/Fをセット又はリセットする手段として下記のType 1~Type 2のいずれかを用い

表 28 において、

P₁ は、メモリの処理すべき領域のディジットをファイルアドレスm_Bとディジットアドレスn_Cで指定する。

P₂ は、P₁で指定されたメモリのディジットの中の所望ビットNに対して0を導入し、Type 2を実行する。

9) メモリの所望領域の1ビットコンディショナルF/Fの内容を判断し、判断結果で次に進むプログラムアドレス(ステップ)を変える手段として下記の手法が用いられる。

表 29 において、

P₁ は、所望のコンディショナルF/Fの1ビットの存在するファイルアドレスm_Bとディジットアドレスn_Cを指定する。

P₂ は、P₁で指定したメモリの領域の中のNで指定するビット(所望のコンディショナルF/Fに対応)の内容が1の場合はP₃をスキップして

ップ)を変える手段として下記の手法が用いられる。

P_1	LB	m_B	n_C
P_2	SKM	N	
P_3	T	P_n	
P_4	OP ₁		
P_n	OP ₂		

表 29

P₄に進みオペレーションOP₁を実行する。もし所望ビットの内容が0の場合は、次のステップP₃に進む。

P₃は、P₂での判断でコンディショナルF/Fが0の時、オペレーションOP₂を実行するため、プログラムステップをP_nに指定する。

10)メモリの所望領域のディジットの内容があらかじめ定められた数値かどうかを判断し、判断結果で次に進むプログラムアドレス(ステップ)

値Nとを比較し、等しい時は、P₄をスキップしてP₅へ進み、オペレーションOP₁を実行する。もし、ACCの内容とNが等しくない時はP₄に進む。

P₄は、プログラムアドレス(ステップ)P_nを指定し、P_nへジャンプする。P_nにてオペレーションOP₂を実行する。

11)メモリの所望領域の複数ディジットの内容が全てあらかじめ定めた数値Nと等しいかどうかを判断し、判断結果で次に進むプログラムアドレス(ステップ)を変える手段として下記の手法が用いられる。

表 31において、

P₁は、判断すべきメモリの領域をファイルアドレスm_Bで指定し、第1のディジットアドレスをn_Eで指定する。

P₂は、比較したい数値NをACCに導入する。

P₃は、ACCの比較値Nとメモリの所望領域の

P ₁	LB	m _B	n _C
P ₂	L	m _B	
P ₃	SKAI	N	
P ₄	T	P _n	
P ₅	OP ₁		
P _n	OP ₂		

表 30

表 30において、

P₁は、判断すべき内容の入っているメモリの領域をファイルアドレスm_Bとディジットアドレスn_Cで指定する。

P₂は、P₁で指定したメモリの内容をACCに導入する。

P₃は、ACCの内容とあらかじめ定められた数

P ₁	LB	m _B	n _E
P ₂	LDI	N	
P ₃	SKAM		
P ₄	T	P _n	
P ₅	ABLI	I	
P ₆	T	P ₃	
P ₇	OP ₁		
P _n	OP ₂		

表 31

所望ディジットの内容とを比較し、一致している時は、続くディジットの比較をするためにP₄をスキップしてP₅へ進む。一致しなかった時はP₄に進む。

P₄は、P₃で不一致の時はすぐオペレーションを実行するためプログラムアドレス(ステップ)をP_nに指定しジャンプさせる。

P_5 は、ディジットアドレスに1を加えることによってディジットアドレスをアップさせる。この処理はメモリの複数ディジットを順次判断していくためのもの。判断してゆくメモリの最終ディジットアドレスをあらかじめ V として決めておくことによって、上記比較を所望ディジット間くり返す。もし途中で不一致状態になれば、 P_4 を経てオペレーション OP_2 を実行するが、 $B_L=V$ になるまで一致し続けた場合には、 P_6 をスキップして P_7 へ進み、オペレーション OP_1 を実行する。 P_6 は、 P_5 にて一致が続く時、 P_3 に戻って判断をくり返す。

12)メモリの所望領域の内容があらかじめ定めた数値 N よりも小さいかどうかを判断し、判断結果で次に進むプログラムアドレス(ステップ)を変える手段として下記の手法が用いられる。

が出るということは2進加算結果が16を超えたことを意味する。つまり $M+(16-N) \geq 16$ であったわけで、これは $M \geq N$ をあらわす。この場合本命令は P_4 をスキップして P_5 に進んでオペレーション OP_1 を実行する。もしキャリイが出なければ $M \geq N$ でなかったわけで、 P_4 に進む。

P_4 は、 $M \geq N$ でない時、このステップでプログラムアドレスを P_n に指定してジャンプし、 P_n でオペレーション OP_2 を実行させる。

18)メモリの所望領域の内容があらかじめ定めた数値 N よりも大きいかどうかを判断し、判断結果で、次に進むプログラムアドレス(ステップ)を変える手段として次の手法が用いられる。

表38において、

P_1 は、判断すべきメモリのファイルアドレス mb とディジットアドレス nc を指定する。

P_2 は、 P_1 で指定したメモリの内容をACCに

P_1	LB	mb	nc
P_2	L		
P_3	ADI		$16-N$
P_4	T		P_n
P_5	OP_1		
P_n	OP_2		

表32

表32において、

P_1 は、判断すべきメモリのファイルアドレス mb とディジットアドレス nc を指定する。

P_2 は、 P_1 で指定したメモリの内容をACCに導入する。

P_3 は、メモリの内容と比較すべき数値を N とすると $(16-N)$ なる数値をオペランドで指定し、その内容とACCのメモリ内容を加算しACCに求める。この加算において第4ビットにキャリ-

導入する。

P_1	LB	mb	nc
P_2	L		
P_3	ADI		$15-N$
P_4	T		P_n
P_5	OP_1		
P_n	OP_2		

表38

P_3 は、メモリの内容と比較する数値を N とする。 $(15-N)$ なる数値をオペランドで指定し、その内容とACCのメモリ内容を加算しACCに求める。この加算で第4ビットにキャリイが出るということは2進加算結果が16を超えたことを意味する。つまり $M+(15-N) \geq 16$ であったわけで、これは $M \geq N+1$ 、すなわち $M > N$ である。この場合、本命令は P_4 をスキップして P_5 に進んでオ

(Type 1)

ベレーションOP₁を実行する。もしキャリーが出なければM>NでないわけでP₄に進む。

P₄は、M>Nでない時、このステップでプログラムアドレス(ステップ)をP_nに指定してジャンプし、P_nでオペレーションOP₂を実行させる。

14)メモリの所望領域の内容を表示する手段として、下記Type 1, Type 2のいずれかの手法が用いられる。

表 8 4 において、

P₁は、表示体を時分割表示させるための桁選択信号を発生させるバッファレジスタWの全内容をリセットするためにWのビット数nをACCに入力する。

P₂は、レジスタWの全内容を1ビット右シフト後、第1ビットに0を入力する。P₀でC₄=1になるまでP₄を介してこれを繰返すことによってWの内容をリセットする。

P₃は、オペランドIAを1111にすることによってACC+1111が実行され、実質的にACC-1を行う。P₁でACCにn₁を入れているのでこの回数を繰返すことによってACC=0になった次の1111との加算の時のみ第4ビットキャリーC₄が0になるので、この時のみP₄へ進み、それ以外はP₅へスキップする。

P₄は、ACC+1111にて第4ビットキャリーC₄=0の時は、Wの全内容を0にしたということで前処理を終え、メモリの表示ステップの第1アドレスP₆をジャンプする。

P₅は、ACC+1111にて第4ビットキャリーC₄=1の時は、まだWの全内容を0にする処理を終えていないので、P₂に戻り、Wへの0入力を繰返す。

P₆は、表示すべき内容の入っているメモリ領域の第1桁をファイルアドレスmAとディジットアドレスnAで指定する。

P₇は、表示用桁選択信号を発生させるレジスタWの内容を1ビット右シフトさせた後、第1ビッ

P ₁	LDI	n ₁
P ₂	WIR	
P ₃	ADI	1111
P ₄	T	P ₆
P ₅	T	P ₂
P ₆	LB	mA nA
P ₇	WIS	
P ₈	LD	mA
P ₉	ATF	
P ₁₀	NPS	
P ₁₁	LDI	n ₂
P ₁₂	ADI	1111
P ₁₃	T	P ₁₆
P ₁₄	T	P ₁₂
P ₁₅	NPR	
P ₁₆	WIR	
P ₁₇	SKBI	nE
P ₁₈	T	P ₈
P ₁₉	SKFA	
P ₂₀	T	P ₆

表 8 4

トに1を入れる。これにて第1桁表示体の桁選択信号供給に備える。

P₈は、指定されたメモリの所望領域の内容をACCに入力する。メモリファイルアドレスは変えずnAである。また、次桁処理に備えディジットアドレスをダウンさせておく。

P₉は、ACCに入っているメモリの内容出力バッファレジスタFに転送する。レジスタFの内容はセグメントデコーダSDに入力させ、セグメント表示用信号を発生させる。

P₁₀は、レジスタWの内容を外部に表示信号として出力するためコンディショナル・フリップフロップNPに1を入れセット状態にする。これに第1桁の表示体でP₉で処理したメモリ内容を表示する。

P₁₁は、1桁分の表示時間を決めるためのカウンタ初期値n₂をACCに入力する。

P₁₂は、P₃と同じように実質的にACC-1を行う。ACCが0になった時はP₁₃に、ACCの内容が0でない時(C₄=1の時)はP₁₄へスキップ

してこの処理を繰返す。

P₁₃は、所望表示時間をP₁₂のACCの内容カウントで処理し、カウントを終了すると、P₁₃を介してP₁₅へジャンプする。このカウント時間が1桁表示時間になる。

P₁₄は、所望表示時間が経過するまでは、P₁₂からP₁₃をスキップしてP₁₄に進み、再びP₁₂にジャンプしこれを繰返す。

P₁₅は、NPをリセットし表示体への桁選択信号の供給をストップする。次にP₁₀で再びNPがセットされるまでは、表示の間接桁信号による重なり表示防止に適用される。

P₁₆は、次桁の表示に備えレジスタWを1ビット右シフトすると共に、第1ビットに0を入れ、実質的に1ビット下位桁にP₇で入力した1をシフトし、次桁選択に備える。

P₁₇は、表示すべきメモリの最終ディジットを終えたかどうかのチェックで、P₈の処理でBL-1がなされているので、(最終ディジット-1)の値がn_gになったかどうかをチェックする。

P₁₈は、最終ディジットが到来していない時はP₈に戻り次桁の表示処理をする。

P₁₉は、例えばフラッグ・フリップフロップF_Aを表示の終了条件とすれば、F_A=1でP₂₀をスキップして一連の表示処理を終える。

P₂₀は、P₁₇でF_A=0ならば再び第1ディジットから表示処理を繰返すべくP₆にジャンプする。

表35において、

P₁は、表示体を時分割表示させるための桁選択信号を発生させるバッファレジスタWの全内容をリセットするために、Wのビット数n₁をACCに入力する。

P₂は、レジスタWの全内容を1ビット右シフト後、第1ビットに0を入力する。P₃でC₄=0になるまでP₅を介してこれをくり返すことによってWの全内容をリセットする。

P₃は、オペランドIAを1111とすることによってACC+1111がなされ、実質的にACC-1を行う。P₁でACCにn₁を入れているのでこの

(Type 2)

P ₁	LDI	n ₁
P ₂	WIR	
P ₃	ADI	1111
P ₄	T	P ₆
P ₅	T	P ₂
P ₆	LB	m _A n _A
P ₇	LD	m _A
P ₈	LXA	
P ₉	LD	m _A
P ₁₀	STPO	
P ₁₁	WIS	
P ₁₂	NPS	
P ₁₃	LDI	n ₂
P ₁₄	ADI	1111
P ₁₅	T	P ₁₇
P ₁₆	T	P ₁₄
P ₁₇	NPR	
P ₁₈	WIR	
P ₁₉	SKBI	
P ₂₀	T	P ₇

表35

回数をくり返すことによってACC=0になった次の1111との加算の時のみ第4ビットキャリーC₄が0になるので、この時のみP₄へ進み、それ以外はP₅へスキップする。

P₄は、ACC+1111にて第4ビットキャリーC₄=0の時は、Wの全内容を0にしたということで前処理を終え、メモリの表示ステップの第1アドレスP₆へジャンプする。

P₆は、ACC+1111にて第4ビットキャリーC₄=1の時は、まだWの全内容を0にする処理を終えてないのでP₂に戻り、Wへの0入力をくり返す。

P₆は、表示すべき内容の入っているメモリ領域の第1位桁の上位4ビットをファイルアドレスm_Aとディジットアドレスn_Aで指定する。

P₇は、指定されたメモリの所望領域の内容をACCに入力する。メモリファイルアドレスは変えずm_Aである。又ディジットアドレスをダウンさせ下位4ビットを指定する。

P₈は、ACCの内容すなわち上位4ビットをデ

ンポラリレジスタXに転送する。

P₉は、指定されたメモリの所望領域の内容をACCに入力する。メモリファイルアドレスは変えずm_Aである。又ディジットアドレスをダウンさせ次桁の上位4ビットを指定する。

P₁₀は、ACCの内容をスタックレジスタSAに、テンポラリレジスタXの内容をスタックレジスタSXに導入する。

P₁₁は、表示用桁選択信号を発生させるレジスタWの内容を1ビット右シフトさせた後、第1ビットに1を入れる。これにて第1桁選択信号供給に備える。

P₁₂は、レジスタWの内容を外部に表示信号として出力するためのコンディショナルF/FNPに1を入れセット状態にする。これにて第1桁の表示体でP₁₀で処理したメモリ内容を表示する。

P₁₃は、1桁分の表示時間を決めるためのカウンタ初期値n₂をACCに入力する。

P₁₄は、P₃と同じ様に実質的にACC-1を行う。ACCが0になった時はP₁₅へ、ACC≠0の時

になったかチェックする。

P₂₀は、最終ディジットが到来していない時はP₇に戻り、次桁の表示処理をする。

15) 押圧されたキースイッチの種類を判別する

(表示中にキー押圧の有無をチェック)ため第36図に示す手法が用いられる。第86図において、P₁~P₁₈は前項14)で説明した表示処理である。

P₁₉は、レジスタWの全ディジットの内容を表示後、フラッグF/FFCをセットし、キー信号I₁~I_nを全て1にする。(第37図参照)

P₂₀は、キー入力KN₁に接続されているキー群のいずれかが押されたかどうかチェックし、押圧されていなければスキップする。

P₂₁は、KN₁に接続されているキー群のいずれかが押されていればP₃₀へジャンプする。

P₂₂~P₂₇は、キー入力KN₂~KF₂の各々に対して、接続されているキー群のいずれかが押されたかどうかを判断し、押されていればP₃₀へジャンプす

(C₄=1の時)はP₁₆へスキップしてこの処理をくり返す。

P₁₅は、所望表示時間をP₁₄のACCの内容カウンタで処理し、カウンタを終了するとP₁₅を介してP₁₇へジャンプする。このカウンタ時間が1桁表示時間になる。

P₁₆は、所望表示時間が経過するまでは、P₁₄からP₁₅をスキップしてP₁₆へ進み、再びP₁₄にジャンプし、これをくり返す。

P₁₇は、NPをリセットし、表示体への桁選択信号の供給をストップする。次にP₁₀で再びNPがセットされるまでは表示の隣接桁信号による重なり表示防止に適用される。

P₁₈は、次桁の表示に備え、レジスタWを1ビット右シフトすると共に第1ビットに0を入れ、実質的に1ビット下位桁にP₇で入力した1をシフトする。

P₁₉は、表示すべきメモリの最終ディジットを終えたかどうかのチェックで、P₉の処理でBL-1がなされているので最終ディジット-1の値n₈

る。

P₂₈は、いずれのキーも押圧されていない場合で、F/FFCをリセットし、キー押圧チェックを終える。

P₂₉は、P₆へジャンプして、再び表示を読める。

P₃₀は、キーが押圧された時にくるステップで、第1のキーストロープ信号I₁発生のためにメモリディジットアドレスを第1状態n₁にする。

P₃₁は、キー入力KN₁に第1キーストロープ信号I₁が入力されたかどうか判断し、入力されていなければP₃₃へスキップする。

P₃₂は、キー入力KN₁に第1キーストロープ信号I₁が入力された時で、キーの種類が判別され、P_Aにジャンプし、この判別されたキーに対応した制御を以下行わせる。そしてそのキー制御を終えた後はP₁へ直接ジャンプして、表示を開始させる。(P₂はP₁へジャンプさせるためのステップ例)

P₃₃~P₃₈は、第1キーストロープ信号I₁に接続されているキーを順次判別、所望キーが押圧され

ていれば $P_B \sim P_D$ へジャンプして、そのキーに対応した制御をする。

P_{39} は、第1キーストロープ信号 I_1 に接続されているキーが押されなかった時で、第2のキーストロープ信号発生のためにメモリディジットアドレスをアップさせる。

$P_{40} \sim P_{44}$ は、所望キーストロープ信号を発生させると共に、 $KN_1 \sim KF_2$ を順次判断し、押圧されたキーの種類を判別し、押圧されたキーに対する制御をするために所望ステップにジャンプする。

P_A は、第1のキーに対する制御ステップである。

P_x は、第1のキー制御完了後 P_1 に戻り表示を再開する。

次に、CPU内のRAM構成について説明する。

第41図にCPU内のRAMマップを示し、第42図にワードメモリのアドレスと記憶内容の対応を示す。

図において、 Y_0, X_0, W_0, Z_0 で示されている $BM_0 \sim BM_3$ までの領域は、主として演算に用い

D：置数時、小数点が指定された状態。

E：置数入力状態。

F：ワードメモリに入力する状態。

G：ワードメモリへの入力時のイニシャル状態。

H：シフト命令が指定された状態。

X_D ：置数時の小数点位置を示す。 X_D は待選処理領域。

$K_U K_L$ ：キーコード、キャラクターコードを記憶。

F_U ：関数コード（ $+ - \times +$ ）を記憶。

ワードメモリ加算カウンタ $U.L$ ：ワードメモリの記憶場所（アドレス）を記憶。

（ワードメモリ加算カウンタ） $U.L$ ：待選処理領域。

Z_S, W_S, X_S, Y_S ：各データレジスタの負号記憶領域。

ランニング表示カウンタ $U.M.L$ ：ランニング表示の移動間隔を決定する為のタイマーカウンタ。

$BM=D \sim BM=F$ は、ワードメモリ領域であり、第42図に示す様な配置で、ここでは24キャラクタ分を確保している。

特開昭58-132849(26)

るレジスタである。 W_0 及び Z_0 のレジスタは前記の表示用キャラクターのバッファメモリとしても用いている。また Z_0 レジスタのうち、 $BL=1$ より $BL=8$ の領域の4ビット目は、 $S O_n$ 出力の出力バッファとして用いられている。（この領域のRAMの情報が直接 $S O_n$ 端子より出力される） V_0 レジスタ（ $BM=4, BL=0 \sim 15$ ）は、データの待選用エリアとして用いられる。数値は、ここでは8桁長を例としている。各レジスタの $BL=4 \sim BL=B$ の領域が仮数部データの記憶に用いられ、 $BL=C$ は、補助用の桁、 $BL=D \sim BL=F$ はデータの重み、すなわち、指数部を記憶する。 $BL=F$ は指数部の負号桁である。 $BM=5$ 及び 6 の領域は各種コンディショナル F/F 及びカウンタを構成する。

図に用いられている記号とその内容は次の通りである。

A： $+ - \times +$ 等の関数キーが押された状態。

B：データが入力された状態。

C： $\sqrt{\quad}$ 等の関数演算が実行された状態。

（本発明実施例の作用説明）

次に、上記実施例の作用をフローチャート等を参照しながら説明する。

第1図から第8図は、その処理手順（フローチャート）を示したもので、第1図は、電源ON状態から、ワードメモリのランニング表示処理を行なうと同時にキー入力を行なう制御部分、第2図は、通常の表示状態（シンボルの点滅表示を含む）におけるキー入力制御部分及び具体的なキー読込み処理及び処理キーコードへの変換を行なう部分、第3図は、読込まれたキーコードを判断してそれぞれの処理部分に分割する部分、あるいは、ワードメモリへのセット状態において、記憶用メモリにキャラクター用のコードを順次書き込む部分、第4図は、 $C L$ キー、 $S E T$ キー、 $C E$ キー、 $C A L L$ キー、 $O F F$ キーの各処理部分、第5図は、ワードメモリへのセット状態において、ワードメモリの内容を表示する為の読出し及び前処理を行なう部分、第6図は、演算結果あるいは、入力された数値を、表示する為の前処理を行なう部分、第7

図は、6ビットのキャラクタコードで準備されている表示内容を、ドットマトリックスのパターン情報に変換するキャラクタジェネレータ(デコーダ)処理部分及び、その情報を外部表示用バッファに出力する部分、第8図は、数値のリードイン(読込み)処理を行なう部分及び、四則(+・-・×・÷)、=等の演算の為の処理を行なう部分である。さらに第43図はキー読込み用のストロブ信号出力とマイクロプロセッサへの入力端子との対応を、第44図は、各キーの内部処理用のキーコード表を、第45図は、ワードメモリに記憶されるキャラクタのコード表を、それぞれ示している。

次に第1図から、第8図までの処理手順を順に説明する。

全体は、大きくいくつかの処理ブロックに分割されて書かれているが、次に示す細かい約束項目によって結合され、大きな一つの処理を構成するようにになっている。

全体は主として、前記した表2乃至表3並びに第36図と共に説明した基本的情報処理の手順

のであるが、処理内容をも併せて記している。

第4図における長方形で囲まれた部分は、一つのまとまった処理であるがかなり大きな処理となる為にそこに挿入せず、全体の流れを乱さない様、第5図、第6図に独立して説明している。

各図において、第46図(Ⅲ)に例示するような矢印で示し、その次に内容が示されているが、この内容は、その部分に続いて処理が継続する行先を示している。すなわち、第46図(Ⅲ)の場合、そこから先は、NOP KEY INPUTと書かれている処理部分に移行することを示している。

第46図(Ⅳ)に示す逆三角形記号は(Ⅲ)図の矢印に対応するものである。例えば(Ⅳ)図に示すフロー開始点は(Ⅲ)図に示す矢印からの処理が移行する。

LB m, n の指定では、mでBMを指定し、nでBLを指定するものとする。例えば、LB 7, Fは、BM=7, BL=FのRAMアドレスを指定するものである。

第1図は、電源投入時より、ワードメモリのランニング表示制御及びキー入力を行なうルーチン

(1)~(Ⅲ)の記載に基いて表現されている。例えばXO-VO (3)という処理は、前記処理リストの(3)と同様の形式で処理されている事を示しており、又例えば第46図(Ⅳ)に例示する図はRというコンディショナルF/Fの状態を判断する処理であり、Y (YES) はセット状態、N (NO) はリセット状態を示しているが、この処理も基本的情報処理(Ⅱ)と同様の形式で処理されていることを示している。(2)-1と示されている場合は基本的情報処理(Ⅱ)のType1の形式で処理されていることを示している。第46図(Ⅴ)に例示する長円形で囲まれている処理は、一つのまとまった処理であり、かつ何度も用いられる処理である為、別途説明している。長円形で示されている処理部分には、別に説明されている同一の処理をあてはめればよい。また第46図(Ⅵ)で示される1及び2は、それぞれ、別途説明されている部分の1, 2の終了条件に対応している。第46図(Ⅶ)に示す破線で囲まれている部分の処理は、(Ⅴ)図の長円形の処理と全く同様であり、そこに示される処理は、何度も用いられるも

である。

20. の処理は、BM, BL=(6, 1), (6, 0)の値が、それぞれ6, 9であるかを判断し、もしそうでない場合には、ワードカウンタリセットとワードメモリクリア処理を行なうものである。ワードメモリ内容は主電源がOFF状態となってもバックアップ(内容保護)されているが、OFFキーによらず、電源を切った場合、あるいは、正常に内容保護されていない場合には、メモリ内のデータが正常である保証がない。これを検出する為、OFFキー操作時(第4図参照) BM, BL=(6, 1)(6, 0)のエリアに6及び9を書き込んだ後、主電源を切る処理を行なっている(00)。メモリが正常に保護されている場合には、電源投入時、OFF時に書き込んだデータがそのまま保護されているから、ワードメモリ内も正常であるという判断を行なっている。したがって、BM, BL=(6, 1)(6, 0)のエリアが6, 9でない場合は、ワードメモリ内も保証されない為、全てのワードメモリ内データをクリアし、ワードメモリを指定するアドレスカウン

タをクリアしている。具体的処理を第3図に記している。

②₁の処理は電源投入時のイニシャライズで、CPU内RAMのBM=7~BM=0の領域を全てクリア(0をセット)している。なお、②₁のREGはその時のメモリーファイルアドレスBMの値で定められるレジスタの意味である。

②₂は、電源投入時の初期設定で、XDに7を、Baに8をセットしている。XDは小数点位置を示し、XD=0の時1桁目(BL=4)、XD=7の時、8桁目(BL=B)に位置することを意味するが、置数時以外は、指数形式で処理する為、小数点は仮数部最上位桁すなわち、XD=7に位置設定される。Baはキーバンス時間を決定する為のバンス処理カウンタで、初期設定に8を指定している。具体的なカウンタの処理は、キー入力時に処理されている(第12図参照)。

②₃からは、ランニング表示の為の前処理である。ワードメモリ内情報の表示であるから(演算結果等のデータの表示でない)、小数点表示用の

内部メモリをクリアし、キャラクタコード準備用内部バッファメモリをもクリアする。(ワードメモリインジケータ判断及びセット)は、ワードメモリに何らかの内容が入力される時は、ワードメモリインジケータ(S)の表示を行なう為の処理である。具体的処理を第4図に示す。この処理は、アドレス(D,0)のデータ、(ワードメモリの先頭の内容)が、0かどうか判断し、0の時は、以降のワードメモリには、入力されていないものと判断出来る為、Sフリップフロップをリセットし、0でない場合には、少なくとも、先頭に何らかの内容が入力されているから、Sフリップフロップをセットしている。

現システム例はディスプレイユニットを9桁表示としている。今ワードメモリ内のキャラクタが、9桁以内かどうかを②₄で判断している。10キャラクタ目の内容が0の場合は、ワードメモリは、9桁以内であるから、スタティック表示(ランニング表示とならない)処理②₅へ移る。②₄内で先ずRAMアドレスをBM,BL:0,2を指定した

LA×14 ATBMによりBM,BL:E,2としているのは、特別な意味はなく直接BM,BL:E,2を指定してもよい。

②₆以降は、ランニング表示処理であり、XD'はキャラクタの表示の切り換わり目に、(スペース)~(スペース)なるキャラクタを、独立して挿入表示させる為の制御用カウンタで初期値は、0に設定する。ランニング表示処理は、第13図-3に示す様にワードメモリBM=Dに記憶されているキャラクタコードを、13図-1に示す形で内部キャラクタコードバッファに準備し(WZ→DSPユニット制御部)の処理で、DSPユニット制御部に出力して表示する。この状態のままで、一定時間間隔表示させる。この時間はRUN DSPカウンタのカウント処理で決定する。一定時間経過後は、こんどは、13図-2に示す形で内部キャラクタコードバッファに次のキャラクタFを準備し、同様に出力し、表示させる。これをくり返すことにより、表示は、順次左へ送られる様になる。(ランニング表示)

13図-1より13図-2への処理は、W及びZメモリを左シフトし、BL=0のメモリに、次のキャラクタコードを入力することで行なう。ワードメモリアドレスカウンタは、次々にBL=0のメモリに準備するキャラクタのアドレスを示すもので、②₇は、これの初期設定を行なっている。

②₈は、走行表示の時間間隔を既定するカウンタの初期設定であり、1キャラクタを処理する都度初期化される。なお②₈のE,8,0は1110-1000-0000を意味している。

②₉において、ワードメモリアドレスカウンタで示されるワードメモリ内の内容を、内部キャラクタコードバッファのBL=0の位置に転送している。具体的処理は第5図に示している。

②₁₀は、内部キャラクタコードバッファに準備されたコードを、外部ディスプレイユニット制御部に転送する処理で、具体的には、第7図に示す。第7図④の処理は、小数点その他シンボル表示情報を、第10図に示す形でBL=9~cに転送するもので、⑤の処理は、内部RAMのBM=2~8の

BL=0~8を制御部側に転送するプログラムである。ディスプレイユニット制御部側へのアドレスは、5ビットで処理している為、ここでは、BMは、F1ビットのみ有効で他は冗長となっている。 $\mathbb{L}_7 \sim \mathbb{L}_{10}$ の一連の処理で、第13図に示す処理を行ない表示をしている。

\mathbb{L}_{11} は、第12図に示すBa時間をカウント処理しながら、キーの入力を判断するプログラムであり、Ba時間を満足した後のキー入力により、各々のキーに対応して、図-4の8ビットのキーコードを作成し、そのコードをKu, K_Lに設定して、第3図の処理に移行する。具体的処理は、第2図に示す。④は、キー入力の有無及びBa時間の処理で、⑤は個々のキーの判断、⑥は、キーコードの変換を行なう。

\mathbb{L}_{12} は、1回のキー入力判断の度に1ずつカウントアップされるカウンタで、カウントオーバーとなるまで、 \mathbb{L}_{11} , \mathbb{L}_{12} の処理をくりかえす。この間の時間が、走行表示の間隔を既定する。 \mathbb{L}_{12} のカウントがオーバーした時点で、次のキャラクタ

(次のキャラクタコードを準備し表示を行なう。)
24キャラクタがフルに入力されており、かつ、前回に24キャラクタ目を出力した場合、 \mathbb{L}_{17} の処理は、25キャラクタ目を指定することになり、OVERとなり2の処理へ移る。この場合、続けて(スペース)₁-(スペース)₂なるキャラクタを強制的に挿入するが、今、 \mathbb{L}_{13} の処理により、1の(スペース)は疑似的に発生されている為、単にX_{D'}のカウンタに4をセットした後、 $\mathbb{L}_{16} \rightarrow \mathbb{L}_{10}$ の処理を行なう。 \mathbb{L}_{13} の1デジット分シフトにより、WとZのレジスタのW(8)、Z(8)は夫々0000となる。

X_{D'}カウンタに2あるいは4が設定されると、 \mathbb{L}_{18} の処理により、NOとなり、 \mathbb{L}_{19} の処理で、X_{D'}カウンタにより、それぞれ、スペースあるいは一キャラクタを内部キャラクタコードバッファに準備する。X_{D'}の2ビットがセットされている時は、キャラクタバッファのBL=0に一キャラクタのコードを準備し、1ビット目が1の時は一の後の(スペース)を準備するのであるが、W, Z

を準備する処理に移る。

\mathbb{L}_{13} で、前回表示したキャラクタコードのシフトを行なう。

\mathbb{L}_{14} は、前回最後に準備したキャラクタコードの上位4ビットが0(そんなキャラクタコードは、入力されることはない。つまり、ワードメモリ内のキャラクタが24キャラクタ未満であり、前回準備したキャラクタは、存在しないもの(スペース)である。)かどうかを判断しており、YESの場合には、X_Dを2に設定し、キャラクタとして一のコードを無条件に準備 \mathbb{L}_{15} して、 \mathbb{L}_{10} の処理に移る。尚 \mathbb{L}_{16} は、走行時間既定用カウンタをプリセットしている。NOの場合 \mathbb{L}_{17} において、次に準備すべきキャラクタのアドレスを指定する為、ワードメモリアドレスカウンタを2カウントアップする。2ずつアップさせるのは、13図-3に示す様にワードメモリ内がBLが2単位で1ワード(キャラクタ)を構成している為である。ワードカウンタが24キャラクタ未満を指定しているときは、1の処理となり、 \mathbb{L}_8 の処理に移る。

のシフトにより、BL=0に00コードが疑似的に発生されている為、そのまま、 \mathbb{L}_{18} の処理にうつる。X_{D'}が0の時は、すでに(スペース)一(スペース)の処理が完了している為、あらためて、ワードメモリの先頭のキャラクタより、表示処理を行なう。 \mathbb{L}_7 以上のX_{D'}カウンタの処理及び表示の関係を第14図に示している。

ワードメモリ内に記憶されているキャラクタコードが9キャラクタ以内の場合には、ワードメモリ内容のスタティック表示を行なう。この処理は、第4図のワードメモリディスプレイよりスタートとする。[ワードメモリDSP前処理]O₁の処理は第5-1図に示すものであり、これはワードカウンタの指定しているキャラクタの内容より順に、前に入力されているキャラクタを、キャラクタコード内部バッファWO, ZOのBL=0よりBL=1, 2...と順に準備する為のものである。先に第5図の処理について説明する。

ワードカウンタは、先頭(最右桁)に表示するキャラクタのメモリアドレスを指しており、その

アドレスをダウンさせながらそれぞれのアドレスの示すキャラクタコードを順次内部キャラクタバッファのBL=0よりアップ方向に入力してゆく(第15図参照)。P₁で内部キャラクタコード内バッファWO,ZOをクリアし、P₃で、前記ワードカウンタの内容を(ワードカウンタ)に転送する。P₄の処理は、(ワードカウンタ)の指定するキャラクタを読み出し、内部キャラクタバッファにおいてcpuXがアドレッシングしているBLのメモリに転送するものである。ここにcpuXとはRAMの桁アドレスを記憶するXBレジスタである。P₂は、cpuXの初期設定であり、0を設定している。この値は、P₄の1回の処理ごとに1ずつカウントアップされている。(内部キャラクタバッファの指定アドレスを1つ次に進めておく。)1つのキャラクタが内部バッファに準備されると、次のキャラクタを準備する為に、キャラクタのアドレスを指定している(ワードカウンタ)をダウン(-2)させる処理がP₅である。終了条件2は、アドレスが1キャラクタ目をさしている時(表示

するキャラクタが9に満たない時)にP₅の処理を行なった時で、(第15図は7桁の例)ここでワードメモリの表示前処理を終了する。終了条件1はワードメモリにまだ準備すべきキャラクタがある場合で、これはワードメモリ内の指定した箇所9桁をスタティック表示する場合の条件である。P₆の処理は、cpuXが9と等しくなったか、すなわち、内部キャラクタバッファの容量が一ぱいになったかを判断しており、YESの場合は、9キャラクタが全て処理された時、NOの場合は、まだ容量に満たない場合で、P₄の処理に戻る。以上の処理で、キャラクタの通常表示準備は終了する。

第1図のL₅よりの説明にもどる。前記第5図では、ワードカウンタの示す位置のキャラクタを基準に表示前処理をするものであり、第1図のL₅以下の処理は、キャラクタの静表示を行なう為に、ワードメモリのどの位置までキャラクタが記憶されているかを検す処理で、終了時には、ワードカウンタは、ワードメモリに記憶している最後のキ

ャラクタのアドレスを指定している。L₅の処理に入った時、ワードメモリ内のキャラクタは9キャラクタ以内であることがわかっている為、L₅では、9キャラクタ目が0かどうかを判断している。[最初にワードカウンタにE0、すなわち、9キャラクタ目のアドレスを設定しておき、そのメモリの値が、0かどうかを判断する]。0でない場合は、今設定されたE0なるアドレスがワードメモリの最終キャラクタのアドレスということになり、ワードメモリ表示処理(前記第5図)に移る。0の時は、8キャラクタ目の内容を次に判断する為に、L₂₀でワードメモリアドレスカウンタのダウン処理を行なう。

L₂₂は、ワードメモリの内容を読み出す処理であるが、LAX Dの命令によりBMは、D(1101)で固定であり、BLは、cpuXで指定される。cpuXの値は、L₂₁において、初期値Eに設定されている。読み出した内容が0でない時は、ワードメモリ表示前処理へ、0の時はL₂₃の処理により、cpuXの値を2カウントダウンして、次のキ

ャラクタを指定する為のアドレスの設定を行なう。以上の処理により、キャラクタコードが0でないキャラクタの位置までワードメモリカウンタをダウンさせることが出来る。この様にして、ワードメモリカウンタの値を設定した後、ワードメモリの静表示を行なう為にワードメモリディスプレイ前処理に移る。以上が、ランニング表示処理の説明である。

キー入力後、すなわち、第2図のKEY IN処理によりキーが読み込まれ、②の処理によりキーコードがKu,KLに発生された後は、第3図の処理に移る。ここでは、各キーに対応して、それぞれの処理に分割する処理あるいは、ワードメモリへのキャラクタの入力状態(SETモード)においては各々のキーに対応したキャラクタコードをワードメモリにセットする処理を行なう。

n₁はエラー状態(Er F/Fがセットされている)において、KEY入力による処理を、CL及びOFFキーのみとする処理で、キーコードが、Ku=0、KL≤2のキーのみがn₂の処理へ移る。

n_2 では K_U が 0 か 1 かでまず分割して、1 の場合は S E T 状態かどうかを判断して、N O の時、0 ~ 9 ・ か + - × ÷ の処理に分割する。

S E T モードの時は、 n_3 の処理で、各々のキーに対応したキーコードに再度変換し、 n_4 以降の処理によりワードメモリに入力する。この時のコードをキャラクタコードで第 4 5 図に示す。
 $K_U=0$ の時、 $K_L \geq 6$ のキーが、S E T モードで指定された時それらは、キャラクタ指定であるから、 n_6 及び n_3 の処理でキャラクタコードに変換して n_4 以降の処理によりワードメモリに入力する。セット状態でない時、 n_7 の処理で K_L の判断によりそれぞれのキーに対応した処理に分割する。
 $G F/F$ は、ワードメモリにキャラクタコードを入力する一番最初の状態であることを示しており、セット状態でない時に押された S E T キーでセットされる(第 4 図)。 n_4 で Y E S となった時は、最初の 1 キャラクタ目の入力の意味しており、 n_6 でワードカウンタをリセットする(ワードメモリの先頭のアドレスを指定)とともに、ワードメモ

リを全てクリアし、 $G F/F$ をリセットし、初期状態を解除する。そして m_0 において、キャラクタコード $K_U K_L$ を、アドレスの指定するワードメモリに貯蔵する。この時、ワードカウンタは、今キャラクタが入力されたワードメモリのアドレスを指定している。次のキャラクタをワードメモリに入力する時は、 $G F/F$ がリセットされている為 n_4-n_9 となり、 n_9 でワードカウンタを 1 キャラクタ分アップ(+2)して今入力しようとするワードメモリのアドレスに指定する。ここで終了条件 2 は、現在すでに 2 4 キャラクタ目を指定している場合のカウンタアップであり、これ以上キャラクタを入力することは出来ない為次の n_{10} の入力処理を行わず無視している。2 4 キャラクタに満たない場合は、 n_{10} において、キャラクタコードをワードメモリに入力する。入力された後ワードメモリ表示以降(第 4 図)の処理で、ワードメモリの表示を行なう。

O_1 は、先に説明した第 5 図の処理を意味しており、表示するキャラクタのキャラクタコードを

内部キャラクタバッファに準備する処理を行なう。この後、 O_2 の処理により、小数点点灯用内部 F/F を全てリセット(キャラクタの表示であり、小数点は点灯しない)し、 O_3 の処理により、ワードメモリにキャラクタが入力されているかどうかを判断し、点灯用 $F/F(S)$ をセットする。 O_4 の処理により内部キャラクタバッファ W Z の情報を、表示ユニット制御部に出力した後、キー入力処理に移る。

キー入力処理は具体的に第 2 図に示す。キー入力処理は、第 1 図に示したランニング表示処理中においても行なわれたが、第 2 図に示すキー入力処理は、キー入力を行なう処理及びキャラクタのセット状態においては、セットモードを示すインジケータの点滅処理のみを行なっている。

m_1 は、点滅時間を設定するカウンタの初期設定を行なっている。カウンタには、ランニング表示カウンタのメモリを用いている。

m_2 の処理においてバランス時間を取りながらキーの読み込みを行なっている。前に説明したよ

うに、キー入力があった場合には、そのまま、第 3 図のキー入力分割処理に移る。キー入力の条件が満足されない場合には、 m_3 の処理に移り、さきほど初期設定したランニング表示カウンタのカウンタアップ処理を行なう。そしてこのカウンタが桁あふれしない場合には、終了条件 1 となり、 m_2 のキー入力処理へもどる。そして m_2 と m_3 の処理をくり返して行ない、ランニング表示カウンタに桁あふれが生じた時、 m_3 の処理において終了条件 2 となり、 m_4 の処理に移る。つまり、 m_2 と m_3 の処理が m_3 の処理で、カウンタの桁あふれが生じるまで、くり返し実行されている間の時間が、 m_4 処理以降のワードメモリインジケータの点滅処理の時間間隔を規定している。

m_4 の処理でワードメモリへのキャラクタの入力状態(セットモード)かどうかを判断している。 $F F/F$ は、セットモード時を示す F/F である。

m_5 の処理において、セットモードにおけるインジケータの点滅処理を行なっている。表示ユニット制御部内の O A (8 ビットアドレス) の内容

の下位1ビットすなわち、Sのビットを読み出し、Sが1の時は、1を引き(リセットする)、0の時は1を加える(セットする)という様にmsの処理の都度、Sのビットを反転する。すなわち、制御部内のキャラクタコードが、デコードされ出力される時、Sに相当する表示セグメントが点滅する。

ランニング表示状態あるいは、通常のキー入力状態において、CL、SET、CE、CALL、OFFの各キーが入力された場合、第8図n7の処理によりそれぞれの処理ルーチンへ移行する。各処理ルーチンを第4図に示す。

<CLキー>

F/Fすなわち、キャラクタのセット状態にあることを示すF/Fの判断をO5で行い、そうでない場合、CLキーは、演算状態あるいはランニング表示状態の解除及びクリアを行なう。O6のCL処理で入力、及び演算用レジスタのクリア及び各種処理用F/Fの初期設定を行なう。〔表示セグメントCL〕は、小数点点灯を示すRAM

セットキーは、セットモードでない場合には、セットモードの指定を行ない、セットモードにおいてはセットモードの解除を行ないCLキー操作と同様の操作を行なうものである。まず、O9の判断を行ない、F/Fがセットされている時はセットモードを意味しているから、O6以降のCL処理を行なう。Fの時は、セットモードを示すF/Fをセットし、さらに表示のクリアを行なう為に、前記のO8以降の処理を行なう。

<CEキー>

CE(クリアエントリ)キーは、セットモードでない場合には、置数のクリアを行ない、セットモードにおいては、シフトキーが指定されていない時には、単にキャラクタ〔X〕を指定し、シフトキーが指定された時には、DELキーとして働く(第19図参照)。

セットモードでない時、O10の処理に移り、B/F/Fがセットされていない時、(置数状態でない時)何も行なわず、O6の表示セグメントCLよりデータの表示処理を行なう。B/F/Fがセッ

のビットを全てクリアする。なお、小数点位置の設定は、次の〔Data DSP前処理〕の処理で行なわれる。この処理は第6図に示すものであり、X0レジスタのデータ内容を表示する形式に変換し、それをキャラクタコードに変換する処理である。詳細は後に述べる。この処理の後、前記O3以降の処理で表示出力を行ない、再びキー入力処理を行なう。セットモード(キャラクタの入力状態)におけるCLキー操作では、ワードメモリのオールクリア及び表示のクリアを行なう。O7の処理でワードメモリのクリア及び、ワードメモリアドレスカウンタのイニシャライズを行なう。O8で、G/F/Fをセットして内部表示用キャラクタバッファ(W0,Z0)をクリアして前記のO2の処理に移行する。G/F/Fは、キャラクタメモリの入力状態の初期状態を意味するF/Fであり、第8図n4の処理により、先頭のキャラクタの入力時のみアドレスカウンタのアップを行なわないようにするためのものである。

<SETキー>

トされている時は、O11の処理で置数状態を示すF/Fをリセットし、現在の置数の前に入力されたデータがV0に記憶されているので、これをX0レジスタに転送し同様にO8の〔表示セグメントCL〕よりデータの表示処理を行なう。これにより、置数状態が解除され以前の状態に復帰する。

セットモードにある時は、O12でシフトキーが指定されたかどうかを判断し、NOの時、すなわち、シフトキーが指定されていない時には、これはキャラクタのXキーであるから、第8図のn3のKu+2-Ku以降の処理に移り、キャラクタコードの入力を行なう。シフトキーが指定されている時には、DEL(デリート)キーとして働く。DELとは、最後に入力されたキャラクタすなわち、表示の一番右端に位置するキャラクタの削除を行なう処理を意味する。O13の処理でキャラクタコードKu.KLを全て0とし、O14でこれらのコードを、現在ワードメモリアドレスカウンタの示すワードメモリに入力する。さらにO15の処理で、ワードカウンタのカウントダウン処理をして

おく。終了条件2は、現在削除されたキャラクタが先頭のキャラクタであった場合（ワードメモリの一番目に入力されていた場合）であり、この時、ワードメモリは全て空である為、現在がワードメモリへの入力の初期状態にあることを示すG F/Fをセットしている。この後、前記の〔ワードメモリ表示前処理〕以降の処理を行ない、ワードメモリの表示を行なう。

〈CALLキー〉

セットモードでない場合には、ワードメモリ内キャラクタのランニング表示を指定する。セットモードにおいては、シフトキーとして動作する。シフトキーは、反転式すなわち、キー操作の都度シフト状態のセットリセットがくり返される。

セットモードでない時はO₁₆の処理により、計算状態のクリアを行ない第1図の③以降のランニング表示処理を行なう。セットモードにおいては、O₁₇の処理により、シフト状態を示すH F/Fの反転処理を行ない、Key入力処理へ続く。

〈OFFキー〉

レジスタに転送し、さらにX_Dの値（小数点の位置を示す値）によってシフトを行ない数値の下4ビットのキャラクタコードを揃える。キャラクタコードの上位4ビットは、数値の場合すべて1（0001）であるから、数値の部分はすべて1を書き込めばよい（第45図参照）。

Q₁によりEがセット（リードイン状態）されている時、Q₂の処理でXOレジスタの内容を第16図-2に示すようにWOレジスタに転送する処理を行なう。この時、Aの状態（四則キーが押されている状態）とĀの状態（四則キーが押される前の状態）により、それぞれ1桁分表示位置が異なる為、図の様な形式及びX_Dの値に設定される。なおX_Dの値は、X_D'にそのまま待避され、処理は全てX_D'の値をもとに行なう。すなわち、XOレジスタの値とX_Dの値は、この〔Data DSP前処理〕の処理では、全く変化せず保持される。

Q₃の処理は、小数点の位置を示すビットをセットし、後に表示ユニット制御部に出力時にそのセグメントを点灯させる為の処理である。X_D'の

OFFキーは、計算機をOFF状態にする。メモリのアドレス(6,0)(6,1)のエリアに9と6を書き込んでおく。これは、ONキーにより電源ONした時に、OFF時のメモリ内容が保護されているかどうかを、簡単に確認する為のデータである。確認はすでに説明した第1図の②において行なっている。OFF動作は、OFF命令によりマイクロプロセッサのハードウェアで処理される。

〔Data DSP前処理〕

データ(XOレジスタ内)をそれぞれの表示様式に変換してさらに、それらの数値を8ビットのキャラクタコードに変換し、内部キャラクタバッファWO, ZOレジスタに揃える処理である。第16図に示すように128.456なる数値がXOレジスタに入力されている場合リードイン中(④)とリードイン状態でない場合(⑤)とでXOレジスタ内及び、X_Dの値が異なる。これらの状態において、第16図の右に示すような表示形式にそれぞれ変換する。基本的な処理としては、XOレジスタの内容をWO

値と小数点の位置との対応は、第16図-3に示す。以上の処理により、数値の下4ビットの設定は終了している。

Q₄は、数値のキャラクタコードの上位4ビットをZOレジスタにセットする為の前処理であり、(Oサプレス処理を含む)、W(8)の内容から順に判断し、その位置のWOレジスタの内容が0でない(数値である。1~9)か、又は、そのアドレス(BL)がX_D'の値と等しくなる(数値0である)最左位置を検索している。

Q₅により数値が負数かどうかを判断し、もし負でない場合にはQ₆の処理によりZOレジスタのQ₄で検索したBLの位置よりBL=0までの位置に1を設定する。もし負の場合には、Q₇の処理によりいま検索したBLのBL+1のWOレジスタの位置に負号[-]の下位4ビットの値Bを設定し、同じBLのZOレジスタの位置よりBL=0までの位置に1を設定する。

Q₈以降の処理は、数値表示とあわせて、現在指定される四則キーのシンボルを表示する処理で、

ある。四則キー(+-×÷)のコードは、第8図(R₁)の処理によりメモリエリアF_uに4ビットコードで入力されている。Q₈ではその四則キーのコードを呼び出しておき、A F/Fすなわち、四則キーが設定されている状態にあるかを判断している。もしAの状態であれば、W Oレジスタ及びZ OレジスタのBL=0の内容をクリアして、この一連の処理を終了する(Q₉)。もしAの状態であれば、B F/F(データの入力状態かあるいは、そうでない状態かを示す)を判断し、もしBであれば、表示の左端BL=8に、 \overline{B} であれば表示の右端BL=0にそれぞれ、四則キーのシンボルを示すコードをW Oレジスタ及びZ Oレジスタに設定し一連の処理を終了する。(第22図参照)

Q₁によりリードイン状態でない場合(四)にはQ₁₁ Q₁₂の処理に移る。

Q₁₁ Q₁₂の処理は、第16図-1に示すような \overline{E} の場合のデータ形式をEの場合のデータ形式に変換する処理を行なっている。Eの場合のデータ形式に変換されれば、あとは、前述のEの場合の

処理と全く同じになる。

Q₁₁の処理は $XO < 1$ 以下の数値(第16図-4参照)の場合(1)の場合)、それを(2)の形式に変換する処理であり、仮数部分をシフトしながら、指数部wを1ずつ加算してゆき、wレジスタが0となるまでくりかえす。

次の $X_D - w_1 \rightarrow X_D'$ は、現在のW Oレジスタにおいてwレジスタの重みを考慮して、仮数部上のどの位置が小数点の位置となるかを求めておりその値を X_D' に転送する処理である。Q₁₂の処理はQ₁₁において変換された形式のデータを、単に右寄せする為の処理であり、W O(4)が0でかつ、 $X_D' \neq 0$ の時(小数点の位置が右端ではない場合)のみ、Wレジスタを右シフトし、 $X_D' - 1 \rightarrow X_D'$ を行なう処理をくりかえす。以上の処理で第16図-1の \overline{E} の形式をEの形式に変換出来た。

Q₁₃の処理は、Q₂の処理とはほぼ同じものであり第16図-2の(1)の形式を(2)及び(3)の形式に変換する処理である。この処理以降は、前記Eの場合のQ₃以降の処理と全く同じ処理を行なえばよ

い。

〈その他〉

第8図に示す処理は、演算に関する処理ルーチンである。

〈0~9.〉は、数値のリードイン処理で、R₃以降は、小数点入力処理であり、R₃で小数点が指定されたことを示すD F/Fをセットし、以降の置数が小数点以下の数値であることを示す。リードインは、X Oレジスタを左シフトし、X₁(BL=4)の位置に数値に相当するコード(4ビット)を入力する方法で行なわれ、X₈まで入力されたかあるいは、X_D(小数点位置)が7(X₈の位置)になるまで行なわれる。R₂は、小数点が押された後の置数時の処理でX左シフトと共に小数点位置も左へ移動(+1)する処理を行なっている。

〈+-×÷〉及び〈=〉は、演算の為の制御を行なう処理ルーチンである。R₁の処理は、四則キーが押された時点で、現在押されたキーが何であるかを4ビットコードで、F_uなる内部メモリエリアに転送しておくものである。

〈Data前処理〉なる処理は、第16図-1に示すような、リードイン状態で入力されているX Oレジスタのデータを、 \overline{E} に示すような形式、すなわち、 $X_D = 7$ に固定(仮数部のデータは $1 \leq X < 10$ の範囲)し、データは、頭づめ(X₈の位置にデータの先頭をあわせる)し、データの重みはXレジスタ部分に設定する形式に変換する処理である。

〈0~9.〉処理においては、データが入力されたということで、B F/F及びE F/Fをセットしてリードイン状態を記憶している。

〈+-×÷〉の処理では四則キーが押された状態を記憶する為、A F/Fをセットしており、同時にデータの入力状態ではなくなった為にB F/Fをリセットしている。

〈=〉キーでは、演算を全て処理し終えた為、B、Aともリセットしている。

次に本発明実施例の変形実施例を説明する。この変形実施例は、第3図に示す変形-1の部分を第88図に示す内容と置換し、第4図に示す変形-2の部分を第89図に示す内容と置換し、第4

図に示す変形-3の部分第40図に示す内容と置換し、第4図において変形-4で示した[NOP KEY INPUT]の処理を実行せず同図において逆三角形記号と共に変形-4と示したところへ移行し、第4図において変形-5で示した[リセットH]を実行しないことにより実施することができる。

この変形実施例が前述の実施例と相違する点は次の通りである。

① ワードメモリにキャラクタを入力中の表示において最右桁にカーソル表示を行ない、次に入力されるキャラクタの位置を示す。ただし、容量がいっぱいになった時には、カーソル表示を行わない。

② シフトキーは、一度指定されると、再度操作されるまで、シフト状態を保持する。この時、シフト状態にあることを示す為、カーソル表示を桁の上に設定する(第17図参照)。

上記①の処理の為、前述の実施例ではワードメモリアドレスカウンタをアップした後にそのアド

いる。

以上の処理にともないDEL(削除処理)が異なる為、変形-2の部分の処理が第89図に示すものとなる。イニシャル状態(C)でない場合で容量OVER状態でない時は、アドレスのダウンを行いその位置Ku Klの8ビットを入力する。いま、OVER状態であれば、この時、前回の入力時、アドレスのアップが行われていない為、アドレスのダウンを行わず、Oコードをワードメモリに入力する。

上記②の処理の為、変形-5においてシフト状態をリセットする処理を削除し、シフト状態の反転処理はO₁₇でのみ行なうものとする。これに伴い、シフト動作のたびにカーソルの表示を変更する必要がある為、変形-4の処理移行先を第4図に示す位置に移している(カーソル表示を行わない場合はNOP KEY INPUTへ移る)。

(本発明の効果)

以上、詳細に説明した本発明の構成及び作用に

レスにキャラクタを入力していた。つまり、入力後、アドレスは現在入力されたキャラクタの位置を示していたが、変形-1の処理では容量がいっぱいである場合を除いてまず現在のアドレスにキャラクタを入力し、その後アドレスをアップさせておく。この時のアドレスは、次に入力されるキャラクタを入力するメモリアドレスがすでに指定されている。n₁₁により容量がいっぱいであるかどうかを判断し、NOの時はm₁₂でキャラクタをワードメモリに貯蔵する。その後、n₁₃の処理でアドレスのアップ操作を行うが、この時、容量がいっぱいの時はカウンタのアップは行わず、OVER F/Fのセットを行う。

カーソル表示を行なう処理は変形-3の処理を追加することにより行なう。セット状態(F状態)でかつOVER F/Fがリセットされている場合にWO及びZOレジスタのBL=0の位置にカーソルのキャラクタコードを入力する。この時、シフトキーが押された状態かどうかによって、カーソルのキャラクタコードが異なる為、O₁₈で処理して

基いて、次に例示するような電子機器を得ることができる。

第21図は本発明による文字表示の経過を示すもので、0.4~0.5秒毎に1桁ずつ表示内容がシフトしてゆく。t₁₀の時1桁目に表示される「H」は文字記憶部の先頭に記憶されている文字であり、t₆の時1桁目に表示される「Y」は文字記憶部に記憶されている文字の最後の文字である。t₁₀で表示される(スペース)→(スペース)は最後の文字「Y」と先頭文字「H」とが連続しないように区切り、識別を安易とするため、自動的に挿入されるものである。尚この例は総てアルファベットであったが数字及び小数点も共に記憶させ表示することができる。また、タイプ印書の都合上、明細書において(一)と印書したものは図面上の▶に相当する。

第22図は計算機として使用した場合のキー操作とそれに伴う表示の例である。S₃で☒を押すれば、被演算数と共に演算記号も表示される。また演算キーの訂正も容易に確認できる。

Sで演算数を置数すれば、被演算数2は消え演算数が表示される。また演算記号の表示位置も最上位桁に移される。

このように演算記号と数値の位置関係により、表示されている数値が演算数なのか被演算数なのか明確に区別することができる。

第23図は表示部の実施例を示す。④は文字記憶の状態表示を行うものでSETモード(文字記憶設定モード)の時点滅、通常モードで文字記憶部に何か文字が記憶されている場合点灯し、通常モードで文字記憶部に何も文字が記憶されていない場合消灯する。なお、④はメモリ、⑤はストレージメモリの表示である。

第20図は本発明実施例の外観正面図であり、(1)は表示部、(2)はキー入力部を示す。各キーのキーシンボルの下段は通常モードのキーを表し、上段はSETモードのキーを表す。SETモードでSHIFTキーを押圧することによって上段の左、上段の右とを切り換えることができる。またSETキーの押圧によりSETモードと通常モードを切

換えることができる。

第18図、第19図はモードに応じてキーシートを交換する場合のキーシンボルを示すものであり、第18図は通常モード、第19図はSETモードを示す。

第24図は本発明実施例の外観斜視図を示すもので、(2)のキーは第18図(3)のキーシートは第19図に対応する。

本発明の効果を列挙すると次の通りである。

- (1) 表示桁数以上の文字記号を記憶、表示することができるため、メモ代りとしてメッセージ等を入力することができる。
- (2) 計算機として使用する場合、演算記号も表示すれば便利であるが本発明によれば特に表示部に演算記号のセグメントを設けず、数値表示用の桁で表示を行うため、より数式に近い表示を行うことができる。
- (3) 記憶した文字記号を繰返し表示する場合、その文字記号の末尾と先頭を続けて表示すれば読み誤る危険性がある。本発明では、記憶している文

字記号の末尾と先頭に特殊記号(スペースも含む)を挿入することができる。

- (4) 記憶している文字記号の数が表示部の表示桁数以上である場合はランニング表示する必要があるが、表示桁数未滿であれば、スタティック(静止)表示する方が読み取り易く、本発明ではいずれをも選択使用することができる。
- (5) 本発明はマイクロプロセッサにより制御するものであるため、文字記号の記憶表示と計算は同種のインストラクションを使用することができ、二つの機能を共に持つ装置を構成することができる。またキー入力部、表示部も大半は共用することができる。
- (6) 記憶されている文字記号の表示は必要に応じてランニング表示することができ、一方、計算機として使用する場合、桁の読み誤りを防止するため、スタティック(静止)表示することができる。
- (7) 文字記号記憶部に何か記憶されているか或は何も記憶されていないかを操作者に指示すること

により、誤って記憶内容を消したりすることもなく、また、第三者に対する伝言を入力しておけば第三者はその指示を見て、記憶内容を呼び出せば伝言を知ることができる。本発明実施例では④セグメントの点灯により記憶部に文字記号が何か記憶されていることを示す。

- (8) 計算機として使用する場合、第2項の演算記号は被演算数の右側に演算数の左側に表示することにより、より数式表示に近づけることができ、計算操作が容易となる。
- (9) 文字記号の書き込み状態であるか否かの区別は表示で行うことが望ましいが、第7項のセグメントを共用することができる。実施例では④セグメントの点滅により、文字記号の書き込み状態を表している。
- (10) 或るメッセージが表示されている状態ですぐに計算を始めることができるため、例えばメッセージ中の数字を使って計算をする場合など便利である。
- (11) 従来、文字記号等を記憶するものは、アルフ

ァベットと数字を区別し、別々のデータとして記憶させるものであった。(これはTEL NOと名前、金額と項目等を対として記憶しておくためのものであった。)本発明はアルファベット(A～Z)と数字(0～9, .)を区別することなく両者とも単なるキャラクタとして記憶させるため、同時に入力することができる。例えば「〇〇〇番まで電話して下さい」といった伝言を入力することもできる。

02) 記憶している文字記号等を表示する場合は通常状態で呼出しキー(CALLキー)を押せばよいが、本発明では電源ON時にも自動的に記憶内容を順次表示するため、誰かに伝言を伝える場合、伝言を受ける者は電源をONするだけでよく、メッセージを入力し送り物とすることもできる。

4. 図面の簡単な説明

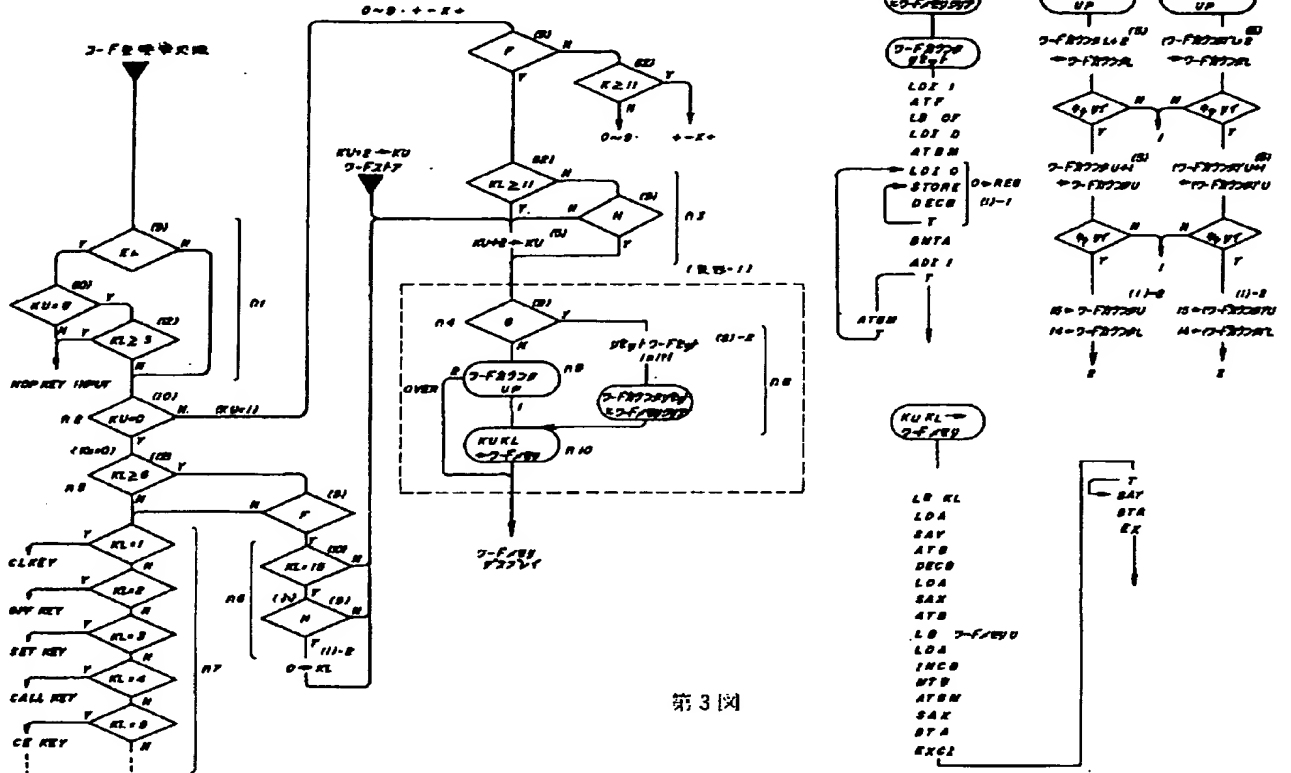
第1図乃至第8図はいずれも本発明実施例の作用を説明するフローチャートである。第9図は本発明実施例の全体構成図である。第10図は本発明実施例の表示ユニット制御部内のRAMのフ

ォートである。第39図は第4図の変形-2の部分の変形を示すフローチャートである。第40図は第4図の変形-3の部分の変形を示すフローチャートである。第41図は本発明実施例のCPU内のRAMマップである。第42図は本発明実施例のワードメモリと記憶内容の対応を示す図である。第43図は本発明実施例のキー読み込み用ストロブ信号出力とマイクロプロセッサへの入力端子との対応を示す図である。第44図は本発明実施例のキーコードを示す図、第45図は本発明実施例のキャラクタコードを示す図である。第46図は第1図乃至第8図のフローチャートの記号の説明図である。

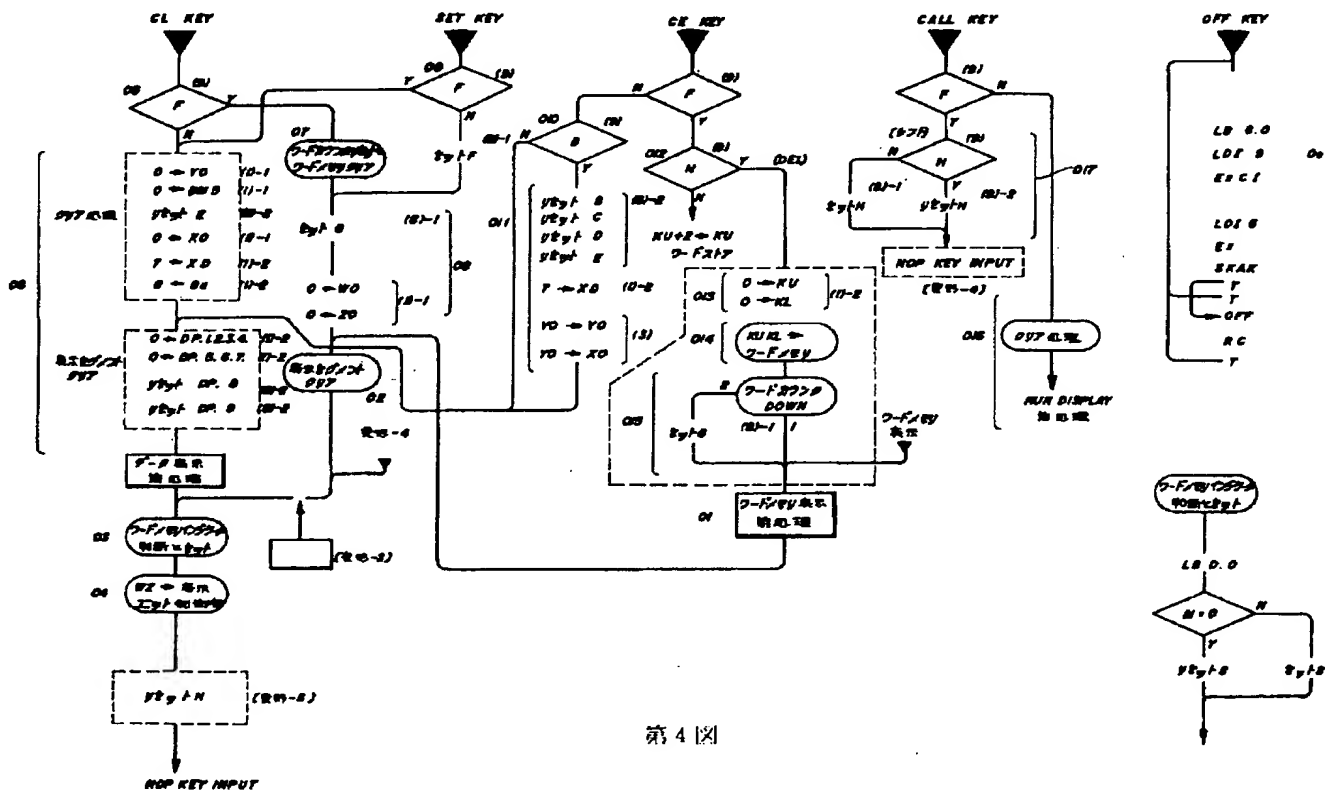
代理人 弁理士 福 士 愛 彦(他2名)

ーマットを示す図、第11図及び第12図は本発明実施例のキー入力時の作用説明図、第13図乃至第17図は本発明実施例の表示の作用説明図である。第18図及び第19図は本発明実施例のモードに応じて交換されるキー配列を示す図である。第20図は本発明実施例の外観正面図である。第21図は本発明実施例による連続的にシフトする表示態様を説明する図、第22図は本発明実施例を計算機として使用したときのキー操作と表示態様を例示する図、第23図は本発明実施例の表示部を示す図、第24図は本発明実施例の外観斜視図である。第25-A図と第25-B図は、本来1枚の図面を2枚に分割して作成したもので、第9図に示す本発明実施例のCPUの具体的回路構成を示す回路図である。第26図乃至第35図は本発明による基本的情報処理の説明図である。第36図はキー入力処理の手順を示すプログラムチャートである。第37図は第36図のチャートを説明するためのキー入力回路図である。第38図は第3図の変形-1の部分の変形を示すフロー

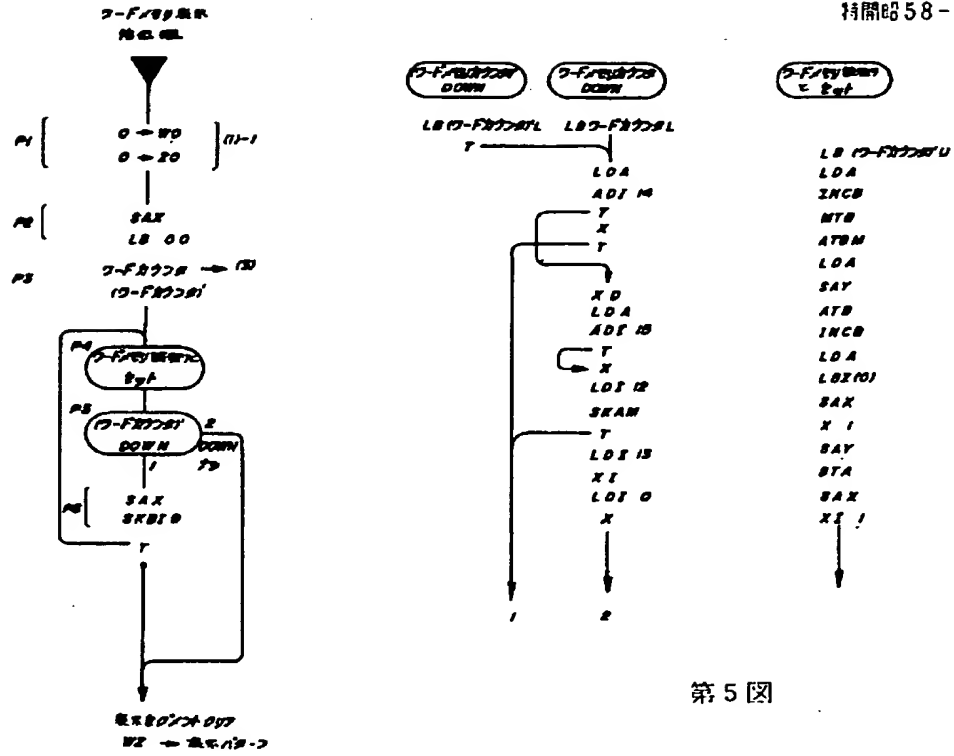




第3図

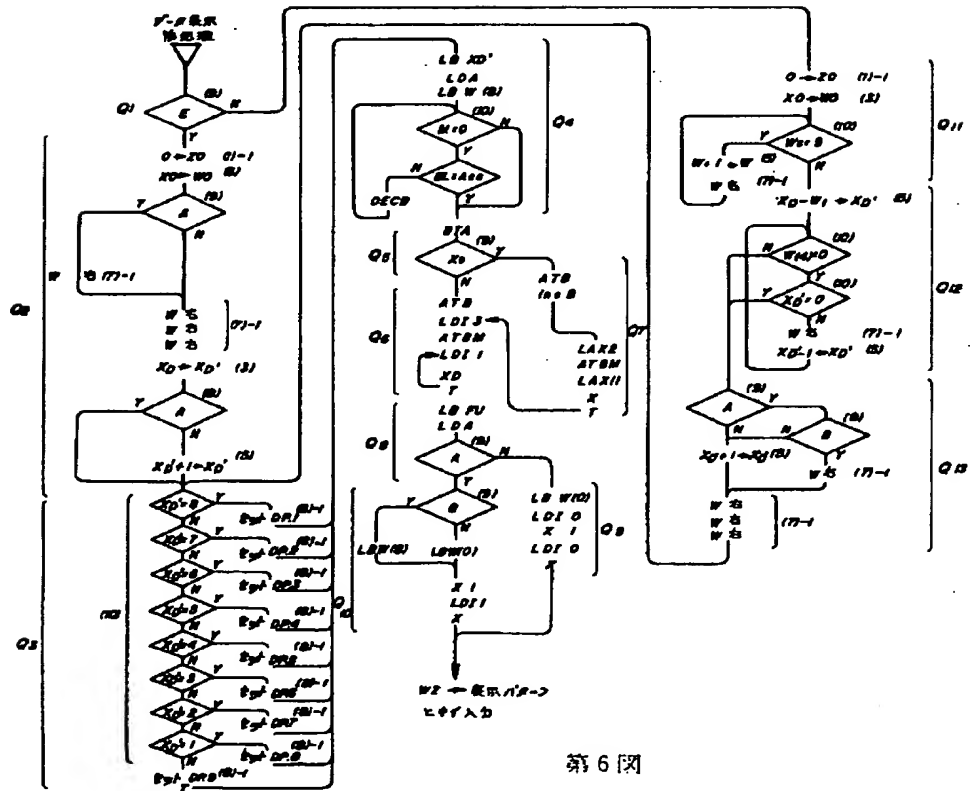


第4図

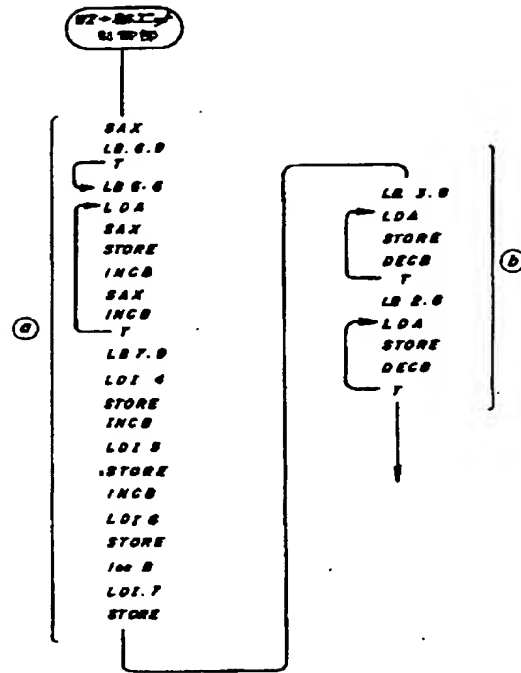


第5-1図

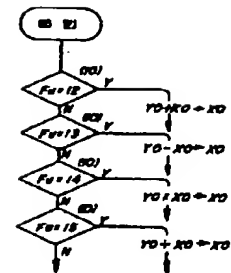
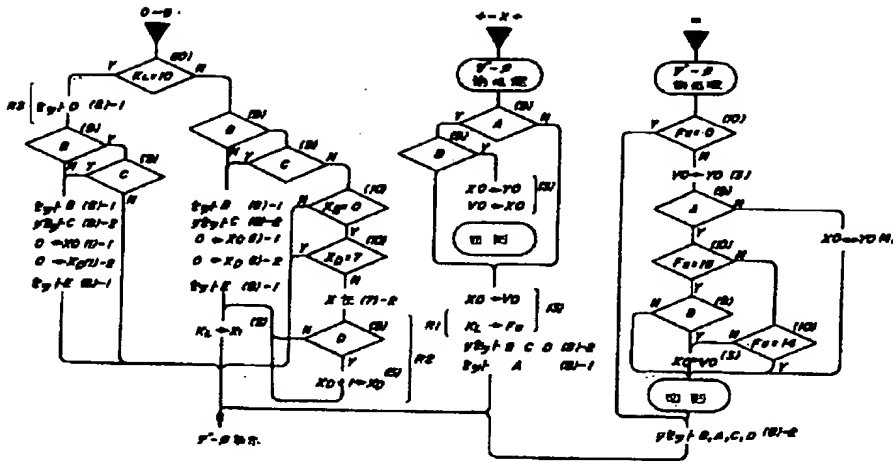
第5図



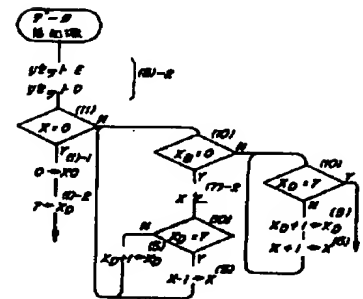
第6図

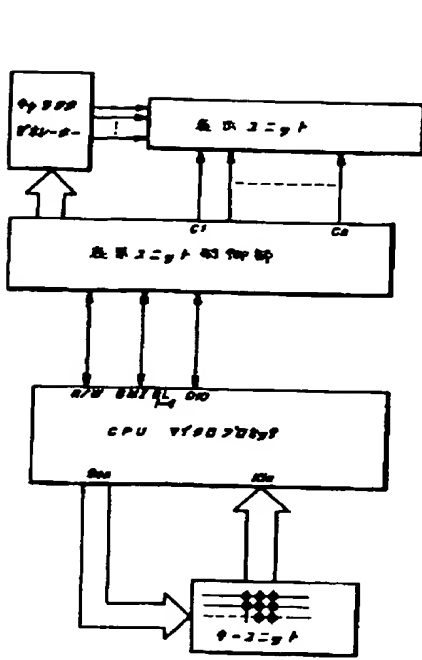


第7図

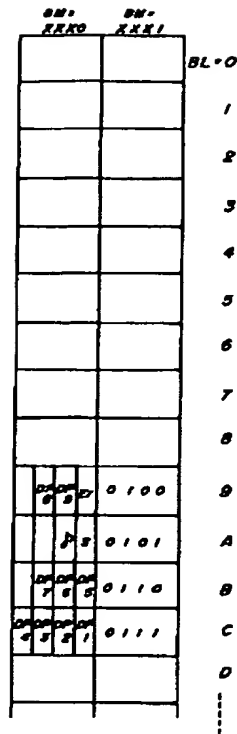


第8図

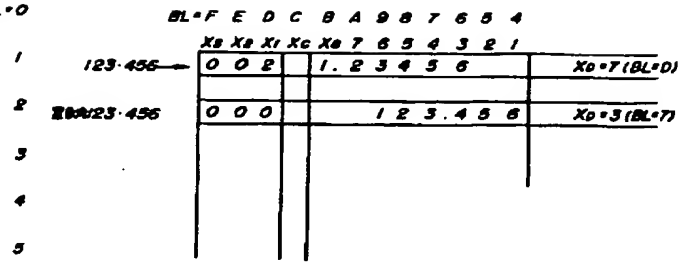




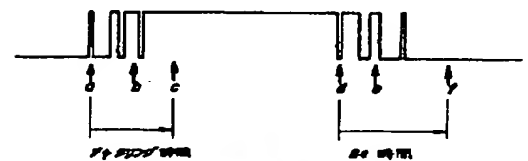
第9図



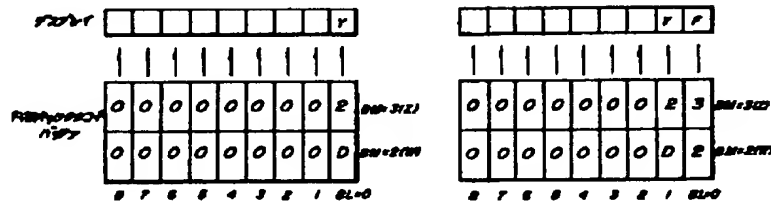
第10図



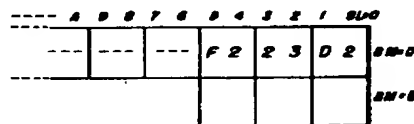
第11図



第12図



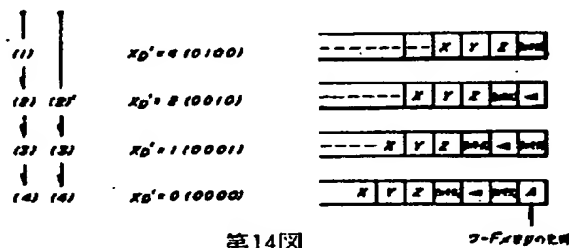
第13図



(3)

CPU 実行部 の場合 の処理
CPU 実行部 の場合 の処理

(1) → (2) → (3) → (4)
(1)' → (2)' → (3)' → (4)'



第14図

7-F, 8-F の処理

OFF CL			
STR	COMP	CALL	SET
$\frac{1}{2}$	$\frac{1}{100}$	$\sqrt{\quad}$	CE
CM	RM	M-	M+
7	8	9	÷
4	5	6	x
1	2	3	-
0	.	=	+

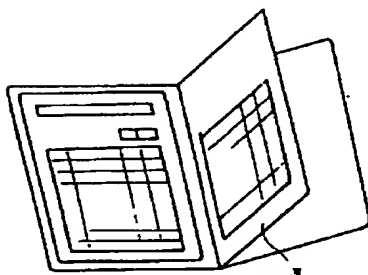
第18図

OFF CL			
Y	Z	SHIFT	SET
SPACE	V	W	DEL
0	H	S	T
7	B	D	P
M	N	O	L
I	J	K	H
E	F	G	C
D	.	=	+

第19図

11	H	A	P	P	Y	B	I	R
12	A	P	P	Y	B	I	R	T
13	P	P	Y	B	I	R	T	H
14	P	Y	B	I	R	T	H	D
15	Y	B	I	R	T	H	D	A
16	B	I	R	T	H	D	A	Y
17	B	I	R	T	H	D	A	Y
18	I	R	T	H	D	A	Y	
19	R	T	H	D	A	Y		
20	T	H	D	A	Y			
21	H	D	A	Y				
22	D	A	Y					

第21図



第24図

OFF CL			
Y	Z	SHIFT	SET
SPACE	V	W	DEL
0	H	S	T
7	B	D	P
M	N	O	L
I	J	K	H
E	F	G	C
D	.	=	+

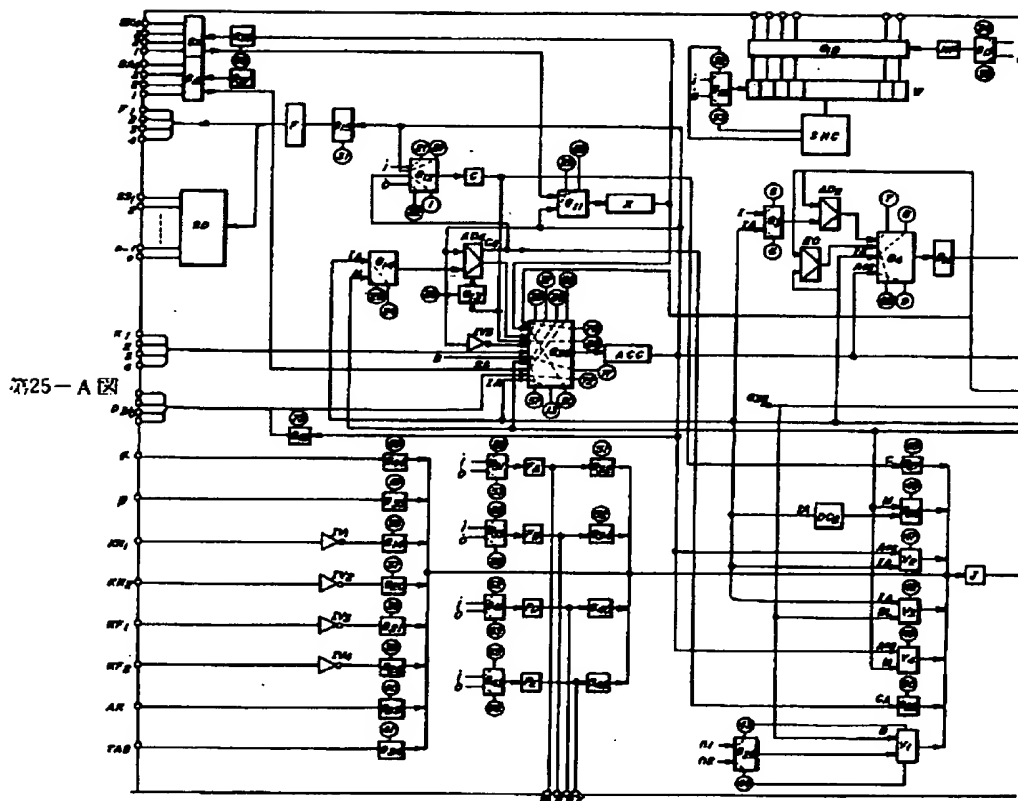
第20図

S1		I.	1
S2		I E.	2
S3		I E. X	3
S4		I E. ÷	4
S5		I E. 0.	5
S6		I E. 0.33	6
S7		3 6.3 6 3 6 3 6	7
S8		3 6.3 6 3 6 3 6 -	8
S9		4 0.	9
S10		- 3.6 3 6 3 6 4 +	0
S11		1 0	1
S12		6.3 6 3 6 3 6	2

第22図

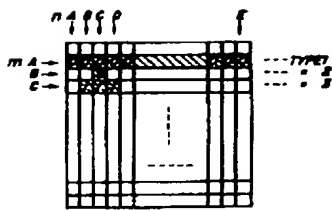
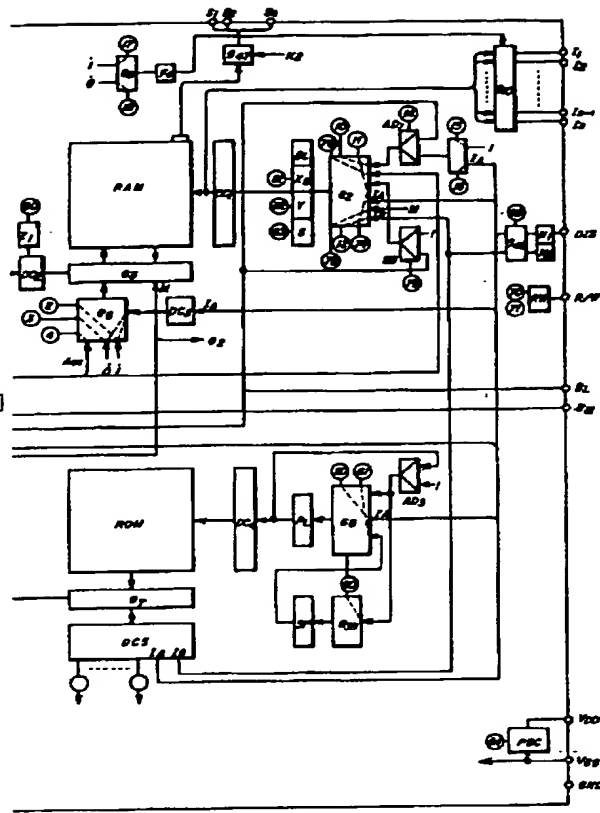
H	A	P	P	Y	B	I	R
---	---	---	---	---	---	---	---

第23図

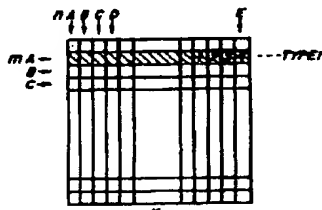


第25-A図

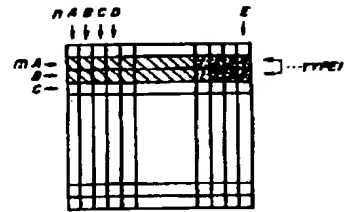
第25-B 図



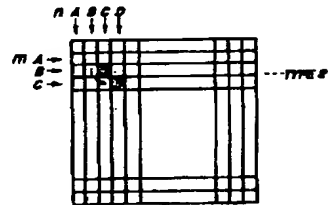
第26図



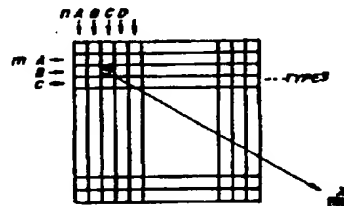
第27図



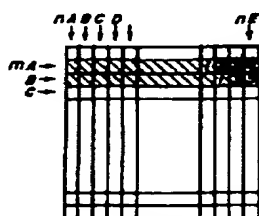
第28図



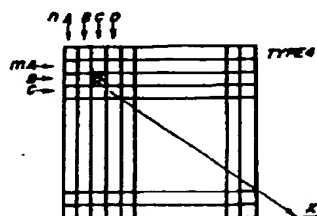
第29図



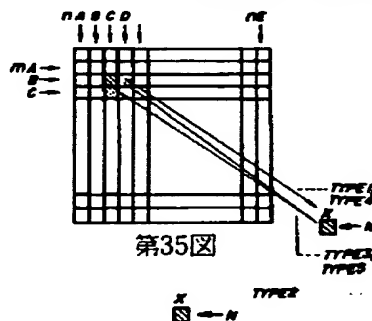
第30図



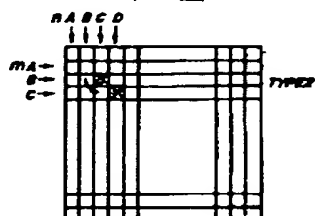
第31図



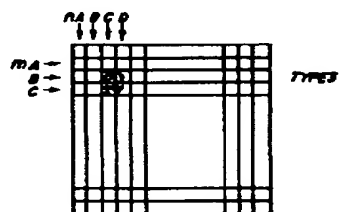
第34図



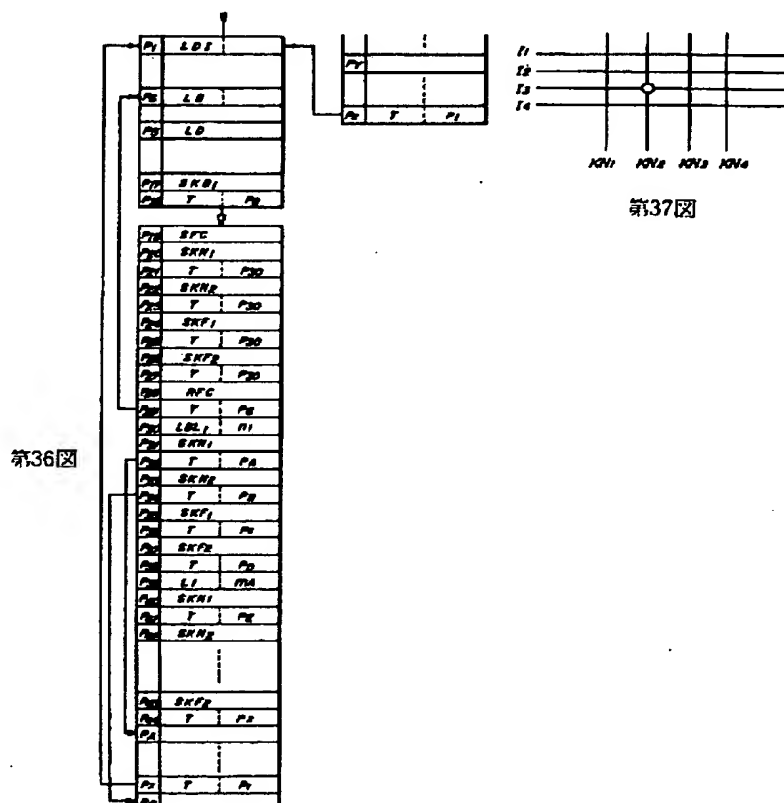
第35図



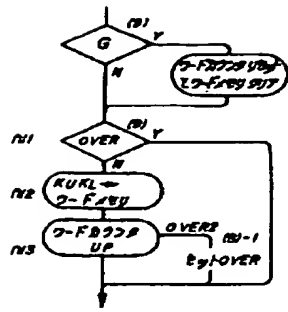
第32図



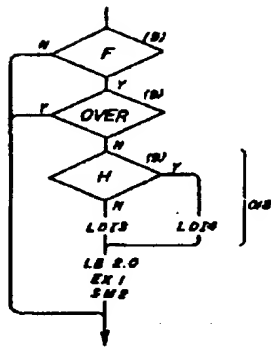
第33図



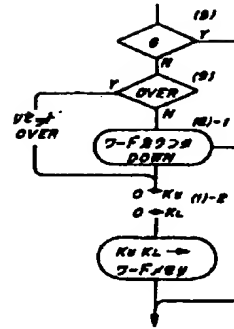
第37图



第38図



第40図



第39図

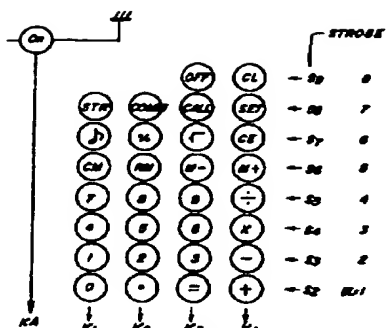
		RM=0000				0001				0010				0011			
DEC 8	BL=0	Y	Y	Y	Y	X	X	X	X	W	W	W	W	X	X	X	X
1																	
2																	
3																	
4		Y	Y	Y	Y	X	X	X	X	W	W	W	W	X	X	X	X
5																	
6		Y	Y	Y	Y	X	X	X	X	W	W	W	W	X	X	X	X
7																	
8																	
9																	
A		Y	Y	Y	Y	X	X	X	X	W	W	W	W	X	X	X	X
B		Y	Y	Y	Y	X	X	X	X	W	W	W	W	X	X	X	X
C		Y	Y	Y	Y	X	X	X	X	W	W	W	W	X	X	X	X
D		Y	Y	Y	Y	X	X	X	X	W	W	W	W	X	X	X	X
E		Y	Y	Y	Y	X	X	X	X	W	W	W	W	X	X	X	X
F		Y	Y	Y	Y	X	X	X	X	W	W	W	W	X	X	X	X

		RM=0000				0001				0010				0011			
DEC 8	BL=0	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1																	
2																	
3																	
4		V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V
5		V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V
6		V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V
7		V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V
8		V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V
9		V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V
A		V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V
B		V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V
C		V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V
D		V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V
E		V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V
F		V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V

第41図

		RM=0000				0001				0010				0011			
DEC 8	BL=0	Y	Y	Y	Y	X	X	X	X	W	W	W	W	X	X	X	X
1																	
2																	
3																	
4		Y	Y	Y	Y	X	X	X	X	W	W	W	W	X	X	X	X
5																	
6		Y	Y	Y	Y	X	X	X	X	W	W	W	W	X	X	X	X
7																	
8																	
9																	
A		Y	Y	Y	Y	X	X	X	X	W	W	W	W	X	X	X	X
B		Y	Y	Y	Y	X	X	X	X	W	W	W	W	X	X	X	X
C		Y	Y	Y	Y	X	X	X	X	W	W	W	W	X	X	X	X
D		Y	Y	Y	Y	X	X	X	X	W	W	W	W	X	X	X	X
E		Y	Y	Y	Y	X	X	X	X	W	W	W	W	X	X	X	X
F		Y	Y	Y	Y	X	X	X	X	W	W	W	W	X	X	X	X

第42図



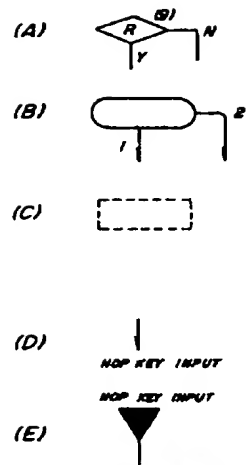
第43図

KL	RL	0000	0001
0000			0
0001	CL	1	
0010	OFF	2	
0011	SET	3	
0100	CALL	4	
0101	CE	5	
0110	CM	6	
0111	RM	7	
1000	M-	8	
1001	M+	9	
1010	-		
1011	=		
1100	%	+	
1101	STR	-	
1110	COMP	X	
1111	♪	+	

第44図

KL	RL	0000	0001	0010	0011	0100	0101	0110	0111
0000	SPACE	0	SPACE	A					
0001		1		E		S	DP.8	DP.1	
0010		2		F	DP.8	DP.8	DP.8	DP.8	
0011		3	2-ND	G		DP.8	DP.8	DP.8	
0100		4	3-ND	I	DP.8	DP.8	DP.8	DP.8	
0101		5	X	J	ETDP8				
0110		6	Q	K					
0111		7	R	M					
1000		8	S	N				DP.8	
1001		9	T	O					
1010		.	W	B					
1011		-	W	C	W				
1100		+	V	D					
1101		-	Y	N					
1110		X	X	L					
1111		+	U	P					

第45図



第46図